MOTOROLA
intelligence everywhere™

digital dna™

MC68HC705C8A
MC68HSC705C8A

Technical Data

M68HC05
Microcontrollers

MC68HC705C8A/D
Rev. 3, 3/2002

WWW.MOTOROLA.COM/SEMICONDUCTORS

MC68HC705C8A
MC68HSC705C8A
Technical Data

To provide the most up-to-date information, the revision of our documents on the World Wide Web will be the most current. Your printed copy may be an earlier revision. To verify you have the latest information available, refer to:

http://www.motorola.com/semiconductors/

The following revision history table summarizes changes contained in this document. For your convenience, the page number designators have been linked to the appropriate location.

Revision History

Date	Revision Level	Description	Page Number(s)
May, 2001	2.1	**1.7 Pin Functions** — Added description of programming voltage (V_{PP}) pin **1.7.2 V_{PP}**	29
		Removed note following **1.7.11 Port D I/O Pins (PD7 and PD5–PD0)**	33
		14.2 Introduction — Updated Motorola contact information	192
March, 2002	3	**14.7 44-Pin Quad Flat Pack (QFP)** — Corrected case outline drawing from Case #824E to Case #824A	195

List of Sections

List of Sections

Table of Contents

Section 1. General Description

Section 2. Memory

Section 3. Central Processor Unit (CPU)

Section 4. Interrupts

Section 5. Resets

Section 6. Low-Power Modes

Section 7. Parallel Input/Output (I/O)

Section 8. Capture/Compare Timer

Section 9. EPROM/OTPROM (PROM)

Section 10. Serial Communications Interface (SCI)

Section 11. Serial Peripheral Interface (SPI)

Section 12. Instruction Set

Section 13. Electrical Specifications

Section 14. Mechanical Specifications

Section 15. Ordering Information

Appendix A. MC68HSC705C8A

Index

List of Figures

List of Figures

Figure	Title	Page

List of Figures

List of Tables

List of Tables

Section 1. General Description

1.1 Contents

1.2 Introduction

The MC68HC705C8A, an enhanced version of the MC68HC705C8, is a member of the low-cost, high-performance M68HC05 Family of 8-bit microcontroller units (MCU). The MC68HSC705C8A, introduced in **Appendix A. MC68HSC705C8A**, is an enhanced, high-speed version of the MC68HC705C8A. The M68HC05 Family is based on the customer-specified integrated circuit (CSIC) design strategy. All MCUs in the family use the M68HC05 central processor unit (CPU) and are available with a variety of subsystems, memory sizes and types, and package types.

1.3 Features

Features of the MC68HC705C8A include:

- M68HC05 central processor unit (CPU)
- On-chip oscillator with crystal/ceramic resonator
- Memory-mapped input/output (I/O)
- Selectable memory configurations
- Selectable programmable and/or non-programmable computer operating properly (COP) watchdog timers
- Selectable port B external interrupt capability
- Clock monitor
- High current drive on pin C7 (PC7)
- 24 bidirectional I/O lines and 7 input-only lines
- Serial communications interface (SCI) system
- Serial peripheral interface (SPI) system
- Bootstrap capability
- Power-saving stop, wait, and data-retention modes
- Single 3.0-volt to 5.5-volt supply (2-volt data-retention mode)
- Fully static operation

- Software-programmable external interrupt sensitivity
- Bidirectional $\overline{\text{RESET}}$ pin

NOTE: *A line over a signal name indicates an active low signal. For example, RESET is active high and $\overline{\text{RESET}}$ is active low. Any reference to voltage, current, or frequency specified in this document will refer to the nominal values. The exact values and their tolerance or limits are specified in* ***Section 13. Electrical Specifications****.*

1.4 Programmable Options

These options are programmable in the mask option registers:

- Enabling of port B pullup devices (see **9.5.2 Mask Option Register 1**)
- Enabling of non-programmable COP watchdog (see **9.5.3 Mask Option Register 2**)

These options are programmable in the option register (see **Figure 1-1**):

- One of four selectable memory configurations
- Programmable read-only memory (PROM) security[1]
- External interrupt sensitivity

Address: $1FDF

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	RAM0	RAM1	0	0	SEC*		IRQ	0
Reset:	0	0	0	0	*	U	1	0

*Implemented as an EPROM cell

= Unimplemented U = Unaffected

Figure 1-1. Option Register (Option)

1. No security feature is absolutely secure. However, Motorola's strategy is to make reading or copying the PROM difficult for unauthorized users.

RAM0 — Random-Access Memory Control Bit 0

1 = Maps 32 bytes of RAM into page zero starting at address $0030. Addresses from $0020 to $002F are reserved. This bit can be read or written at any time, allowing memory configuration to be changed during program execution.

0 = Provides 48 bytes of PROM at location $0020–$005F.

RAM1 — Random-Access Memory Control Bit 1

1 = Maps 96 bytes of RAM into page one starting at address $0100. This bit can be read or written at any time, allowing memory configuration to be changed during program execution.

0 = Provides 96 bytes of PROM at location $0100.

SEC — Security Bit

This bit is implemented as an erasable, programmable read-only memory (EPROM) cell and is not affected by reset.

1 = Bootloader disabled; MCU operates only in single-chip mode

0 = Security off; bootloader can be enabled

IRQ — Interrupt Request Pin Sensitivity Bit

IRQ is set only by reset, but can be cleared by software. This bit can be written only once.

1 = $\overline{\text{IRQ}}$ pin is both negative edge- and level-sensitive.

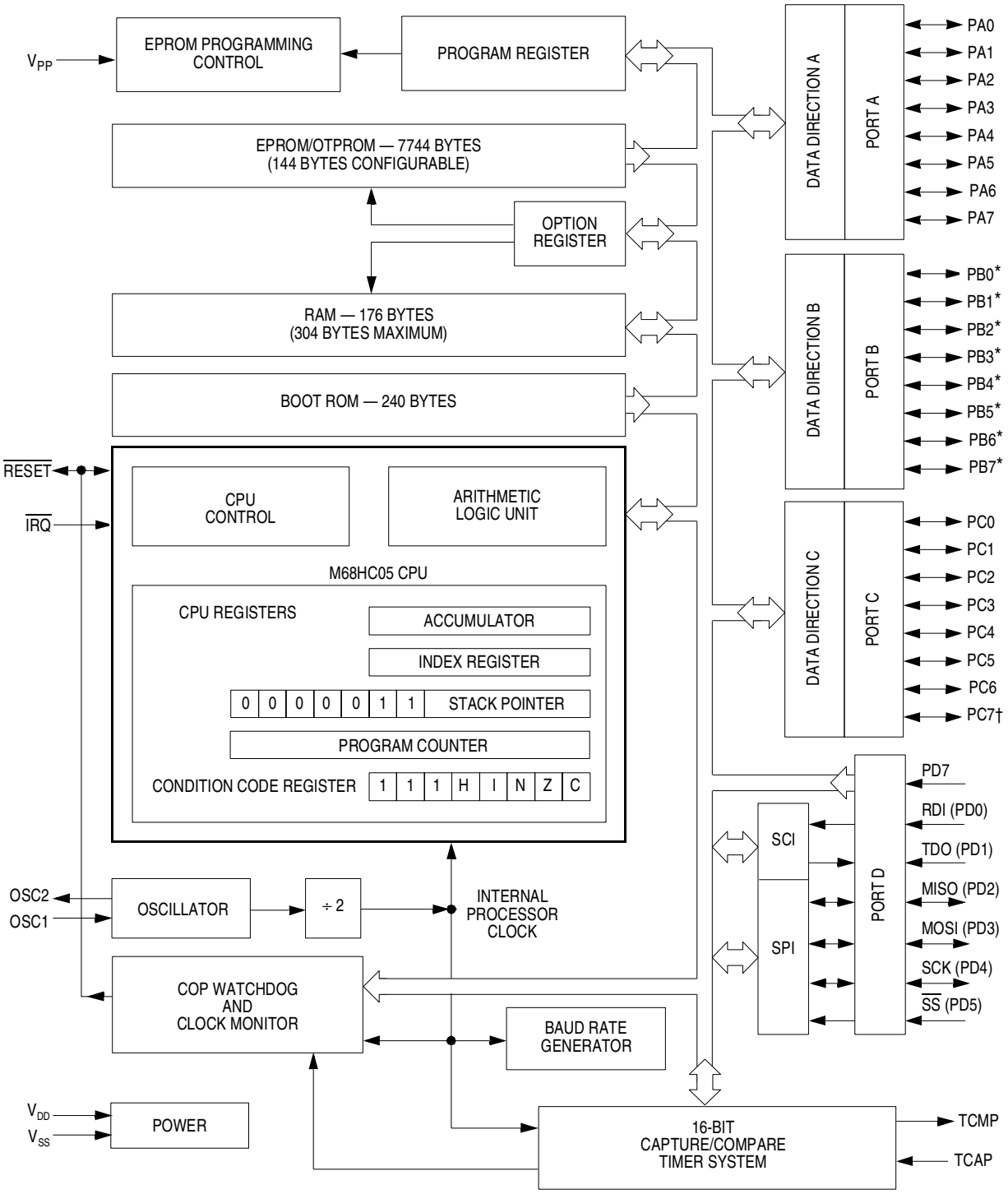
0 = $\overline{\text{IRQ}}$ pin is negative edge-sensitive only.

Bits 5, 4, and 0 — Not used; always read 0

Bit 2 — Unaffected by reset; reads either 1 or 0

1.5 Block Diagram

Figure 1-2 shows the structure of the MC68HC705C8A.



* Port B pins also function as external interrupts.
† PC7 has a high current sink and source capability.

Figure 1-2. MC68HC705C8A Block Diagram

1.6 Pin Assignments

The MC68HC705C8A is available in six packages:

- 40-pin plastic dual in-line package (PDIP)
- 40-pin ceramic dual in-line package (cerdip)
- 44-lead plastic-leaded chip carrier (PLCC)
- 44-lead ceramic-leaded chip carrier (CLCC)
- 44-pin quad flat pack (QFP)
- 42-pin shrink dual in-line package (SDIP)

The pin assignments for these packages are shown in **Figure 1-3**, **Figure 1-4**, **Figure 1-5**, and **Figure 1-6**.

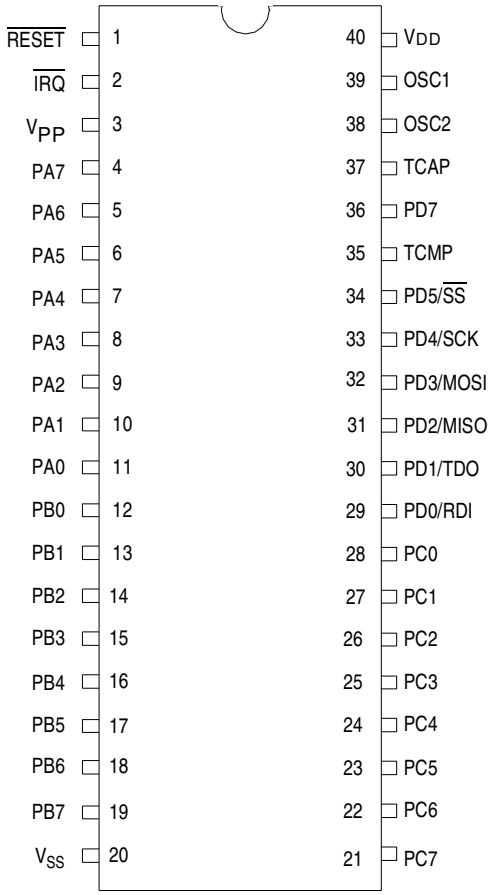


Figure 1-3. 40-Pin PDIP/Cerdip Pin Assignments

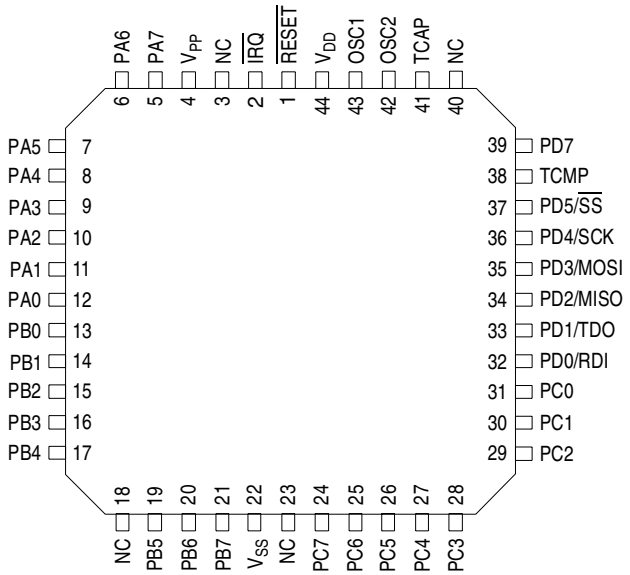


Figure 1-4. 44-Lead PLCC/CLCC Pin Assignments

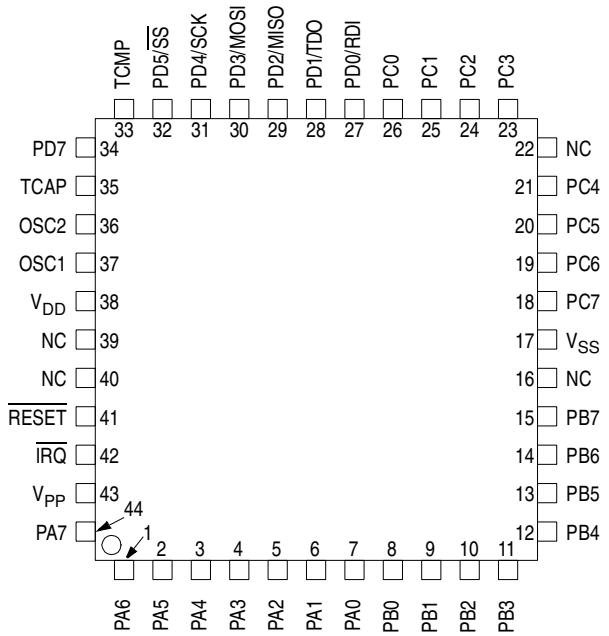


Figure 1-5. 44-Pin QFP Pin Assignments

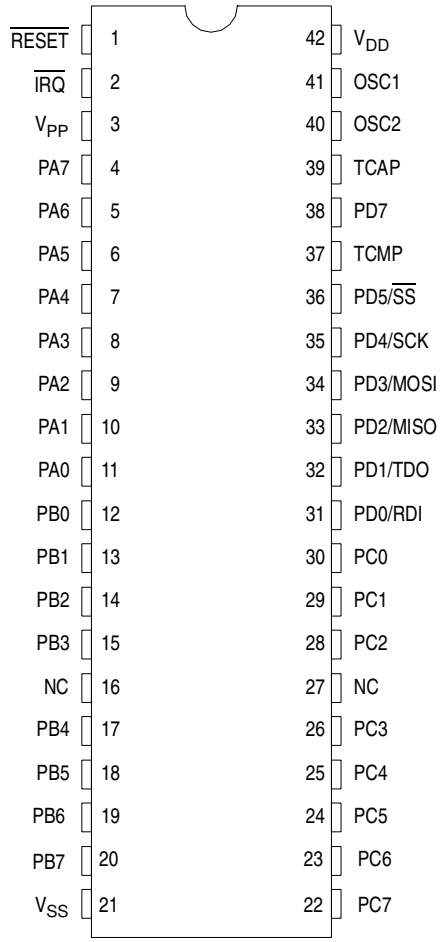


Figure 1-6. 42-Pin SDIP Pin Assignments

1.7 Pin Functions

This subsection describes the MC68HC705C8A signals. Reference is made, where applicable, to other sections that contain more detail about the function being performed.

1.7.1 V_{DD} and V_{SS}

V_{DD} and V_{SS} are the power supply and ground pins. The MCU operates from a single power supply.

Very fast signal transitions occur on the MCU pins, placing high short-duration current demands on the power supply. To prevent noise problems, take special care to provide good power supply bypassing at the MCU. Place bypass capacitors as close to the MCU as possible, as shown in Figure 1-7.

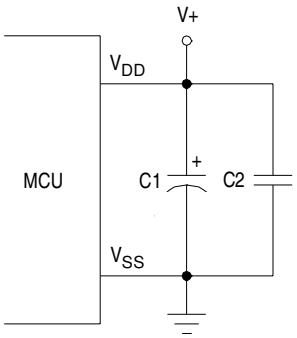


Figure 1-7. Bypassing Layout Recommendation

1.7.2 V_{PP}

This pin provides the programming voltage to the EPROM array. For normal operation, V_{PP} shuld be tied to V_{DD}.

NOTE: *Connecting the V_{PP} pin (programming voltage) to V_{SS} (ground) could result in damage to the MCU.*

1.7.3 OSC1 and OSC2

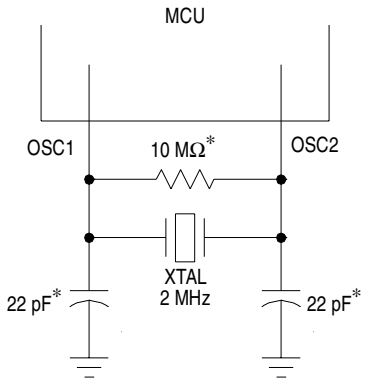
The OSC1 and OSC2 pins are the control connections for the 2-pin on-chip oscillator. The oscillator can be driven by:

- Crystal resonator
- Ceramic resonator
- External clock signal

NOTE: *The frequency of the internal oscillator is f_{OSC}. The MCU divides the internal oscillator output by two to produce the internal clock with a frequency of f_{OP}.*

1.7.3.1 Crystal Resonator

The circuit in **Figure 1-8** shows a crystal oscillator circuit for an AT-cut, parallel resonant crystal. Follow the crystal supplier's recommendations, because the crystal parameters determine the external component values required to provide reliable startup and maximum stability. The load capacitance values used in the oscillator circuit design should account for all stray layout capacitances. To minimize output distortion, mount the crystal and capacitors as close as possible to the pins.



*Starting value only. Follow crystal supplier's recommendations regarding component values that will provide reliable startup and maximum stability.

Figure 1-8. Crystal Connections

NOTE: *Use an AT-cut crystal and not a strip or tuning fork crystal. The MCU might overdrive or have the incorrect characteristic impedance for a strip or tuning fork crystal.*

1.7.3.2 Ceramic Resonator

To reduce cost, use a ceramic resonator instead of a crystal. Use the circuit shown in **Figure 1-9** for a 2-pin ceramic resonator or the circuit shown in **Figure 1-10** for a 3-pin ceramic resonator, and follow the resonator manufacturer's recommendations.

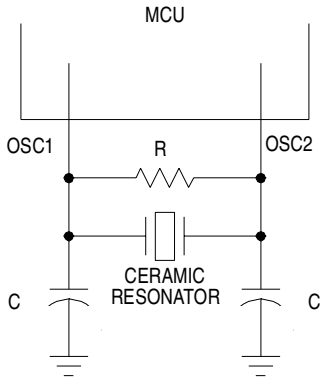


Figure 1-9. 2-Pin Ceramic Resonator Connections

The external component values required for maximum stability and reliable starting depend upon the resonator parameters. The load capacitance values used in the oscillator circuit design should include all stray layout capacitances. To minimize output distortion, mount the resonator and capacitors as close as possible to the pins.

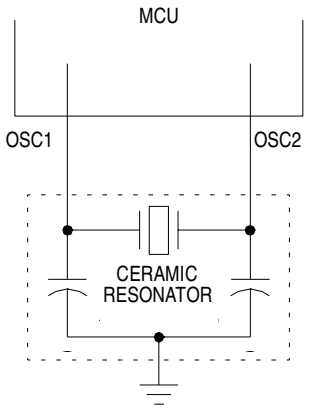


Figure 1-10. 3-Pin Ceramic Resonator Connections

NOTE: *The bus frequency (f_{OP}) is one-half the external or crystal frequency (f_{OSC}), while the processor clock cycle (t_{CYC}) is two times the f_{OSC} period.*

1.7.3.3 External Clock Signal

An external clock from another CMOS-compatible device can drive the OSC1 input, with the OSC2 pin unconnected, as **Figure 1-11** shows.

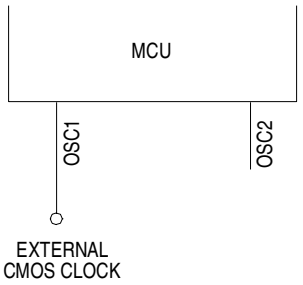


Figure 1-11. External Clock

NOTE: *The bus frequency (f_{OP}) is one-half the external frequency (f_{OSC}) while the processor clock cycle is two times the f_{OSC} period.*

1.7.4 External Reset Pin ($\overline{\text{RESET}}$)

A logic 0 on the bidirectional $\overline{\text{RESET}}$ pin forces the MCU to a known startup state. The $\overline{\text{RESET}}$ pin contains an internal Schmitt trigger as part of its input to improve noise immunity. See **Section 5. Resets**.

1.7.5 External Interrupt Request Pin ($\overline{\text{IRQ}}$)

The $\overline{\text{IRQ}}$ pin is an asynchronous external interrupt pin. The $\overline{\text{IRQ}}$ pin contains an internal Schmitt trigger as part of its input to improve noise immunity. See **4.3.2 External Interrupt (IRQ)**.

1.7.6 Input Capture Pin (TCAP)

The TCAP pin is the input capture pin for the on-chip capture/compare timer. The TCAP pin contains an internal Schmitt trigger as part of its input to improve noise immunity. See **Section 8. Capture/Compare Timer**.

1.7.7 Output Compare Pin (TCMP)

The TCMP pin is the output compare pin for the on-chip capture/compare timer. See **Section 8. Capture/Compare Timer**.

1.7.8 Port A I/O Pins (PA7–PA0)

These eight I/O lines comprise port A, a general-purpose, bidirectional I/O port. The pins are programmable as either inputs or outputs under software control of the data direction registers. See **7.3 Port A**.

1.7.9 Port B I/O Pins (PB7–PB0)

These eight I/O pins comprise port B, a general-purpose, bidirectional I/O port. The pins are programmable as either inputs or outputs under software control of the data direction registers. Port B pins also can be configured to function as external interrupts. See **7.4 Port B**.

1.7.10 Port C I/O Pins (PC7–PC0)

These eight I/O pins comprise port C, a general-purpose, bidirectional I/O port. The pins are programmable as either inputs or outputs under software control of the data direction registers. PC7 has a high current sink and source capability. See **7.5 Port C**.

1.7.11 Port D I/O Pins (PD7 and PD5–PD0)

These seven lines comprise port D, a fixed input port. All special functions that are enabled (SPI and SCI) affect this port. See **7.6 Port D**.

Section 2. Memory

2.1 Contents

2.2 Introduction

This section describes the organization of the on-chip memory.

2.3 Memory Map

The central processor unit (CPU) can address eight Kbytes of memory and input/output (I/O) registers. The program counter typically advances one address at a time through memory, reading the program instructions and data. The programmable read-only memory (PROM) portion of memory — either one-time programmable read-only memory (OTPROM) or erasable, programmable read-only memory (EPROM) — holds the program instructions, fixed data, user-defined vectors, and interrupt service routines. The random-access memory (RAM) portion of memory holds variable data.

I/O registers are memory-mapped so that the CPU can access their locations in the same way that it accesses all other memory locations. The shared stack area is used during processing of an interrupt or

subroutine call to save the CPU state. The stack pointer decrements during pushes and increments during pulls.

Figure 2-1 is a memory map of the MCU. Addresses $0000–$001F, shown in **Figure 2-2**, contain most of the control, status, and data registers. Additional I/O registers have these addresses:

- $1FDF, option register
- $1FF0, mask option register 1 (MOR1)
- $1FF1, mask option register 2 (MOR2)

2.4 Input/Output (I/O)

The first 32 addresses of memory space, from $0000 to $001F, are the I/O section. These are the addresses of the I/O control registers, status registers, and data registers. See **Figure 2-2** for more information.

2.5 RAM

One of four selectable memory configurations is selected by the state of the RAM1 and RAM0 bits in the option register located at $1FDF. Reset or power-on reset (POR) clears these bits, automatically selecting the first memory configuration as shown in **Table 2-1**. See **9.5.1 Option Register**.

Table 2-1. Memory Configurations

RAM0	RAM1	RAM Bytes	PROM Bytes
0	0	176	7744
1	0	208	7696
0	1	272	7648
1	1	304	7600

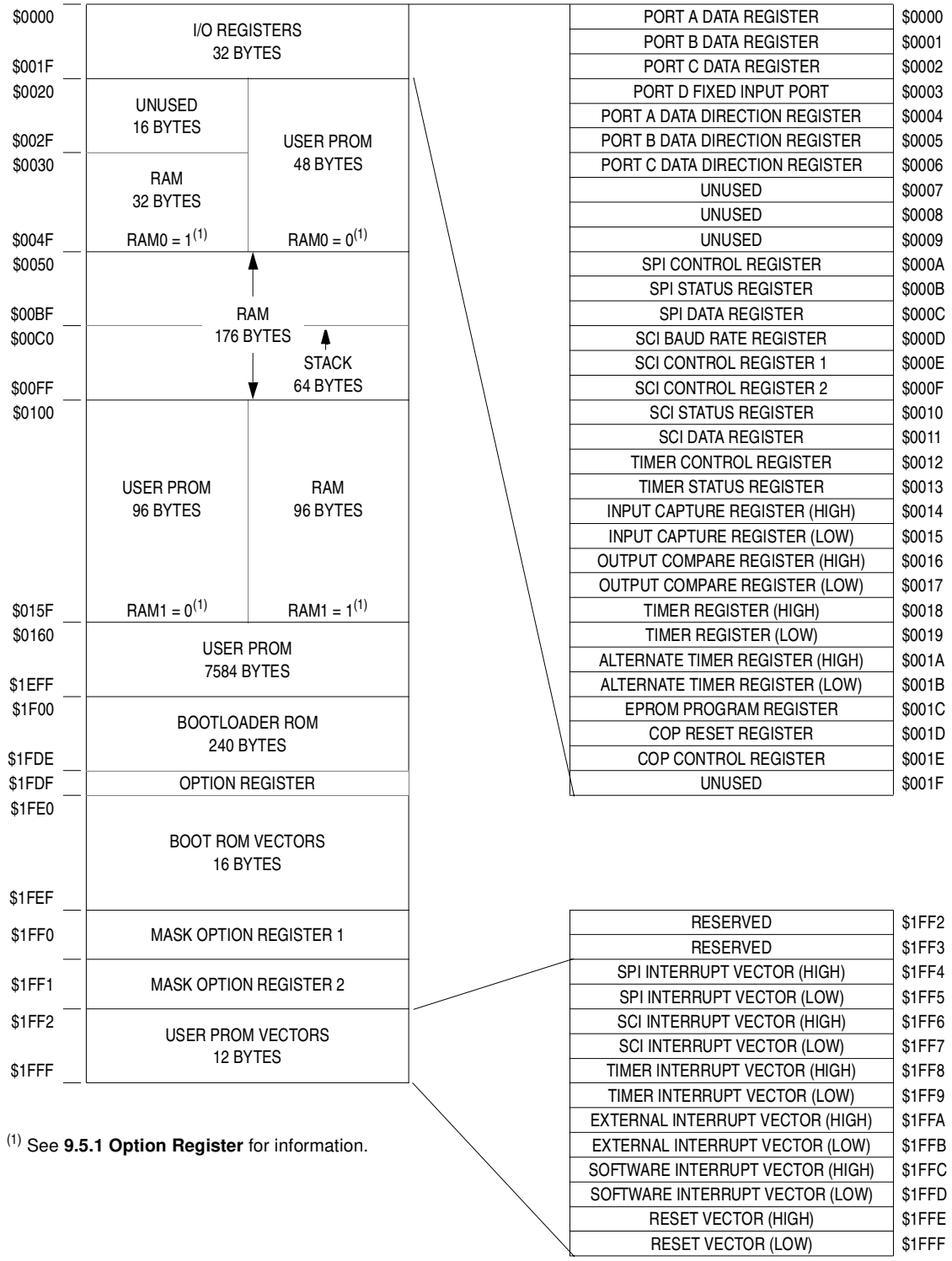
NOTE: *Be careful when using nested subroutines or multiple interrupt levels. The CPU can overwrite data in the stack RAM during a subroutine or during the interrupt stacking operation.*

2.6 EPROM/OTPROM (PROM)

An MCU with a quartz window has a maximum of 7744 bytes of EPROM. The quartz window allows the EPROM erasure with ultraviolet light. In an MCU without a quartz window, the EPROM cannot be erased and serves a maximum 7744 bytes of OTPROM (see **Table 2-1**). See **Section 9. EPROM/OTPROM (PROM)**.

2.7 Bootloader ROM

The 240 bytes at addresses $1F00–$1FEF are reserved ROM addresses that contain the instructions for the bootloader functions. See **Section 9. EPROM/OTPROM (PROM)**.



(1) See **9.5.1 Option Register** for information.

Figure 2-1. Memory Map

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0000	Port A Data Register (PORTA) See page 78.	Read: Write:	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
		Reset:	Unaffected by reset							
$0001	Port B Data Register (PORTB) See page 81.	Read: Write:	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
		Reset:	Unaffected by reset							
$0002	Port C Data Register (PORTC) See page 85.	Read: Write:	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
		Reset:	Unaffected by reset							
$0003	Port D Fixed Input Register (PORTD) See page 88.	Read:	PD7		SS	SCK	MOSI	MISO	TDO	RDI
		Write:								
		Reset:	Unaffected by reset							
$0004	Port A Data Direction Register (DDRA) See page 79.	Read: Write:	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
		Reset:	0	0	0	0	0	0	0	0
$0005	Port B Data Direction Register (DDRB) See page 82.	Read: Write:	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
		Reset:	0	0	0	0	0	0	0	0
$0006	Port C Data Direction (DDRC) See page 86.	Read: Write:	DDRC7	DDRC6	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0
		Reset:	0	0	0	0	0	0	0	0
$0007	Unimplemented									
$0008	Unimplemented									
$0009	Unimplemented									

= Unimplemented U = Unaffected

Figure 2-2. I/O Register Summary (Sheet 1 of 4)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$000A	SPI Control Register (SPCR) See page 149.	Read:	SPIE	SPE		MSTR	CPOL	CPHA	SPR1	SPR0
		Write:								
		Reset:	0	0		0	U	U	U	U
$000B	SPI Status Register (SPSR) See page 151.	Read:	SPIF	WCOL		MODF				
		Write:								
		Reset:	0	0		0				
$000C	SPI Data Register (SPDR) See page 149.	Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Blt 2	Bit 1	Bit 0
		Write:								
		Reset:	Unaffected by reset							
$000D	Baud Rate Register (Baud) See page 136.	Read:			SCP1	SCP0		SCR2	SCR1	SCR0
		Write:								
		Reset:	U	U	0	0	U	U	U	U
$000E	SCI Control Register 1 (SCCR1) See page 130.	Read:	R8	T8		M	WAKE			
		Write:								
		Reset:	U	U		U	U			
$000F	SCI Control Register 2 (SCCR2) See page 131.	Read:	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0010	SCI Status Register (SCSR) See page 133.	Read:	TDRE	TC	RDRF	IDLE	OR	NF	FE	
		Write:								
		Reset:	1	1	0	0	0	0	0	U
$0011	SCI Data Register (SCDR) See page 129.	Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Write:								
		Reset:	Unaffected by reset							
$0012	Timer Control Register (TCR) See page 94.	Read:	ICIE	OCIE	TOIE	0	0	0	IEDG	OLVL
		Write:								
		Reset:	0	0	0	0	0	0	U	0

= Unimplemented U = Unaffected

Figure 2-2. I/O Register Summary (Sheet 2 of 4)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0013	Timer Status Register (TSR) See page 96.	Read:	ICF	OCF	TOF	0	0	0	0	0
		Write:								
		Reset:	U	U	U	0	0	0	0	0
$0014	Input Capture Register High (ICRH) See page 100.	Read:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
		Write:								
		Reset:	Unaffected by reset							
$0015	Input Capture Register Low (ICRL) See page 100.	Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Write:								
		Reset:	Unaffected by reset							
$0016	Output Compare Register High (OCRH) See page 101.	Read:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
		Write:								
		Reset:	Unaffected by reset							
$0017	Output Compare Register Low (OCRL) See page 101.	Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Write:								
		Reset:	Unaffected by reset							
$0018	Timer Register High (TRH) See page 97.	Read:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
		Write:								
		Reset:	Reset initializes TRH to $FF							
$0019	Timer Register Low (TRL) See page 97.	Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Write:								
		Reset:	Reset initializes TRL to $FC							
$001A	Alternate Timer Register High (ATRH) See page 99.	Read:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
		Write:								
		Reset:	Reset initializes ATRH to $FF							
$001B	Alternate Timer Register Low (ATRL) See page 99.	Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Write:								
		Reset:	Reset initializes ATRL to $FC							

= Unimplemented U = Unaffected

Figure 2-2. I/O Register Summary (Sheet 3 of 4)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$001C	EPROM Programming Register (PROG) See page 109.	Read:	0	0	0	0	0	LAT	0	PGM
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$001D	Programmable COP Reset Register (COPRST) See page 64.	Read:								
		Write:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Reset:	U	U	U	U	U	U	U	U
$001E	Programmable COP Control Register (COPCR) See page 64.	Read:	0	0	0	COPF	CME	PCOPE	CM1	CM0
		Write:								
		Reset:	0	0	0	U	0	0	0	0
$001F	Unimplemented									
$1FDF	Option Register (Option) See page 116.	Read:	RAM0	RAM1	0	0	SEC*		IRQ	0
		Write:								
		Reset:	0	0	0	0	*	U	1	0
$1FF0	Mask Option Register 1 (MOR1) See page 117.	Read:	PBPU7	PBPU6	PBPU5	PBPU4	PBPU3	PBPU2	PBPU1	PBPU0/ COPC
		Write:								
		Reset:	Unaffected by reset							
$1FF1	Mask Option Register 2 (MOR2) See page 118.	Read:								NCOPE
		Write:								
		Reset:	Unaffected by reset							

*Implemented as an EPROM cell

= Unimplemented U = Unaffected

Figure 2-2. I/O Register Summary (Sheet 4 of 4)

Section 3. Central Processor Unit (CPU)

3.1 Contents

3.2 Introduction

This section describes the central processor unit (CPU) registers.

3.3 CPU Registers

Figure 3-1 shows the five CPU registers. These are hard-wired registers within the CPU and are not part of the memory map.

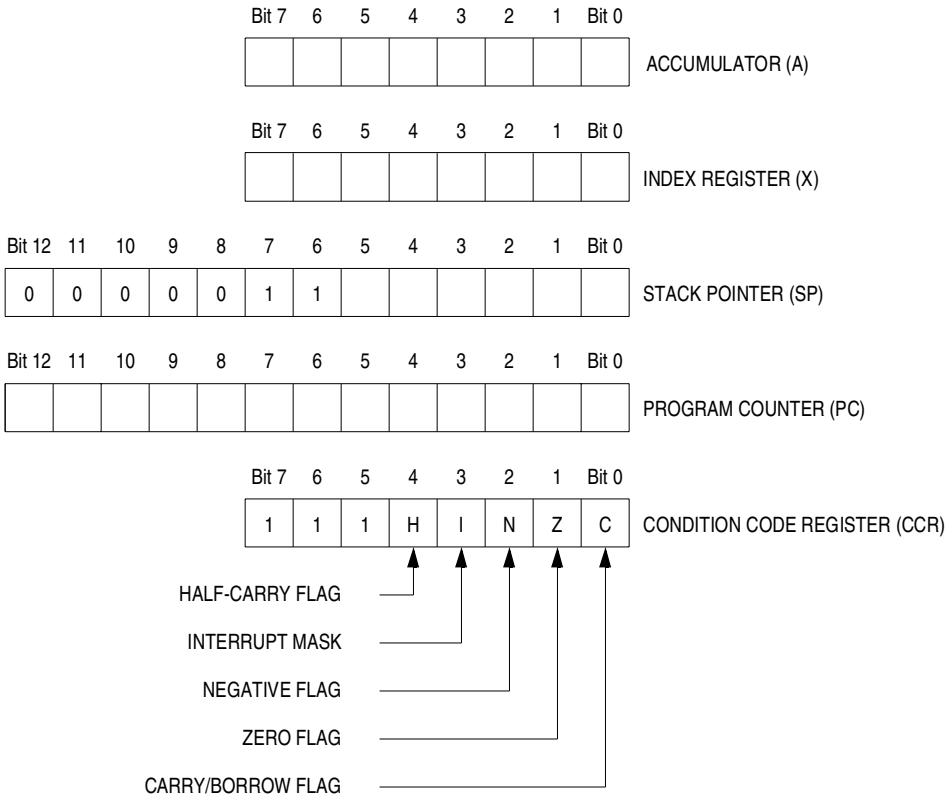


Figure 3-1. Programming Model

3.3.1 Accumulator

The accumulator (A) shown in **Figure 3-2** is a general-purpose 8-bit register. The CPU uses the accumulator to hold operands and results of arithmetic and non-arithmetic operations.

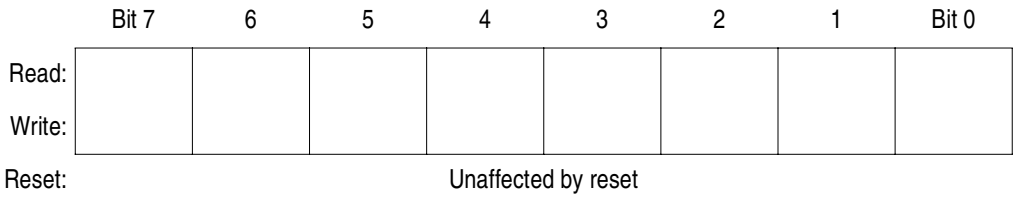


Figure 3-2. Accumulator (A)

3.3.2 Index Register

In the indexed addressing modes, the CPU uses the byte in the index register (X) shown in **Figure 3-3** to determine the conditional address of the operand. See **12.3.5 Indexed, No Offset**, **12.3.6 Indexed, 8-Bit Offset**, and **12.3.7 Indexed, 16-Bit Offset** for more information on indexed addressing.

The 8-bit index register also can serve as a temporary data storage location.

Figure 3-3. Index Register (X)

3.3.3 Stack Pointer

The stack pointer (SP) shown in **Figure 3-4** is a 13-bit register that contains the address of the next free location on the stack. During a reset or after the reset stack pointer (RSP) instruction, the stack pointer initializes to $00FF. The address in the stack pointer decrements as data is pushed onto the stack and increments as data is pulled from the stack.

The seven most significant bits of the stack pointer are fixed permanently at 0000011, so the stack pointer produces addresses from $00C0 to $00FF. If subroutines and interrupts use more than 64 stack locations, the stack pointer wraps around to address $00FF and begins writing over the previously stored data. A subroutine uses two stack locations. An interrupt uses five locations.

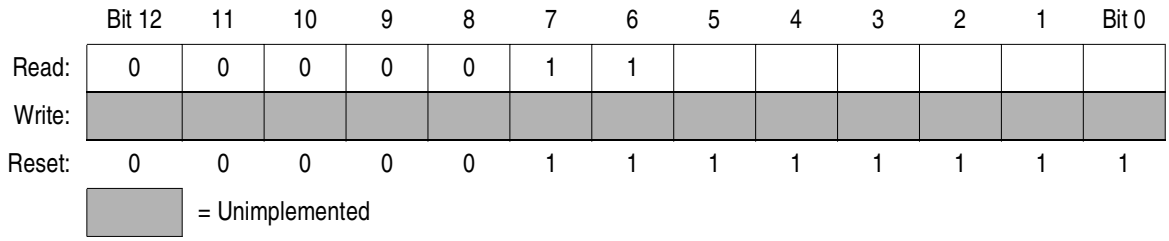


Figure 3-4. Stack Pointer (SP)

3.3.4 Program Counter

The program counter (PC) shown in **Figure 3-5** is a 13-bit register that contains the address of the next instruction or operand to be fetched.

Normally, the address in the program counter automatically increments to the next sequential memory location every time an instruction or operand is fetched. Jump, branch, and interrupt operations load the program counter with an address other than that of the next sequential location.

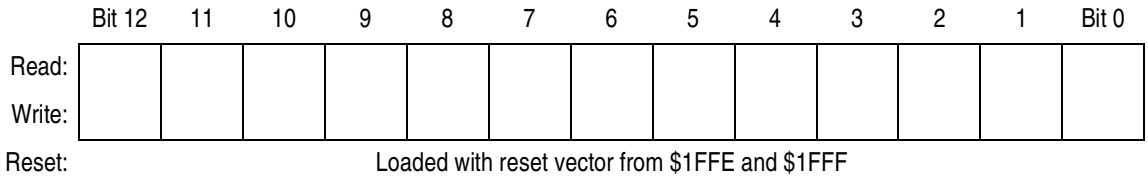


Figure 3-5. Program Counter (PC)

3.3.5 Condition Code Register

The condition code register (CCR) shown in **Figure 3-6** is an 8-bit register whose three most significant bits are permanently fixed at 111. The condition code register contains the interrupt mask and four bits that indicate the results of prior instructions.

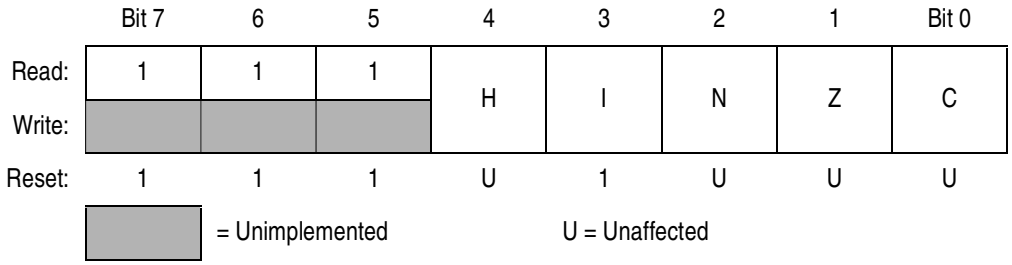


Figure 3-6. Condition Code Register (CCR)

H — Half-Carry Bit

The CPU sets the half-carry flag when a carry occurs between bits 3 and 4 of the accumulator during an add without carry (ADD) or add with carry (ADC) operation. The half-carry bit is required for binary-coded decimal (BCD) arithmetic operations. Reset has no affect on the half-carry flag.

I — Interrupt Mask Bit

Setting the interrupt mask (I) disables interrupts. If an interrupt request occurs while the interrupt mask is a logic 0, the CPU saves the CPU registers on the stack, sets the interrupt mask, and then fetches the interrupt vector. If an interrupt request occurs while the interrupt mask is set, the interrupt request is latched. The CPU processes the latched interrupt as soon as the interrupt mask is cleared again.

A return-from-interrupt (RTI) instruction pulls the CPU registers from the stack, restoring the interrupt mask to its cleared state. After a reset, the interrupt mask is set and can be cleared only by a CLI, STOP, or WAIT instruction.

N — Negative Flag

The CPU sets the negative flag when an arithmetic operation, logical operation, or data manipulation produces a negative result (bit 7 in the results is a logic 1). Reset has no effect on the negative flag.

Z — Zero Flag

The CPU sets the zero flag when an arithmetic operation, logical operation, or data manipulation produces a result of $00. Reset has no effect on the zero flag.

C — Carry/Borrow Flag

The CPU sets the carry/borrow flag when an addition operation produces a carry out of bit 7 of the accumulator or when a subtraction operation requires a borrow. Some logical operations and data manipulation instructions also clear or set the carry/borrow bit. Reset has no effect on the carry/borrow flag.

3.4 Arithmetic/Logic Unit (ALU)

The arithmetic/logic unit (ALU) performs the arithmetic and logical operations defined by the instruction set. The binary arithmetic circuits decode instructions and set up the ALU for the selected operation. Most binary arithmetic is based on the addition algorithm, carrying out subtraction as negative addition. Multiplication is not performed as a discrete operation but as a chain of addition and shift operations within the ALU. The multiply instruction requires 11 internal clock cycles to complete this chain of operations.

Section 4. Interrupts

4.1 Contents

4.2 Introduction

This section describes how interrupts temporarily change the normal processing sequence.

4.3 Interrupt Sources

These sources can generate interrupts:

- Software instructions (SWI)
- External interrupt pin ($\overline{\text{IRQ}}$)
- Port B pins
- Serial communications interface (SCI):
  - SCI transmit data register empty
  - SCI transmission complete
  - SCI receive data register full
  - SCI receiver overrun
  - SCI receiver input idle
- Serial peripheral interface (SPI):
  - SPI transmission complete
  - SPI mode fault
  - SPI overrun

The $\overline{\text{IRQ}}$ pin, port B pins, SCI, and SPI can be masked (disabled) by setting the I bit of the condition code register (CCR). The software interrupt (SWI) instruction is non-maskable.

An interrupt temporarily changes the program sequence to process a particular event. An interrupt does not stop the execution of the instruction in progress but takes effect when the current instruction completes its execution. Interrupt processing automatically saves the central processor unit (CPU) registers on the stack and loads the program counter with a user-defined vector address.

4.3.1 Software Interrupt

The software interrupt instruction (SWI) causes a non-maskable interrupt.

4.3.2 External Interrupt ($\overline{IRQ}$)

An interrupt signal on the $\overline{IRQ}$ pin latches an external interrupt request. After completing the current instruction, the CPU tests these bits:

- IRQ latch
- I bit in the CCR

Setting the I bit in the CCR disables external interrupts.

If the IRQ latch is set and the I bit is clear, the CPU then begins the interrupt sequence. The CPU clears the IRQ latch while it fetches the interrupt vector, so that another external interrupt request can be latched during the interrupt service routine. As soon as the I bit is cleared during the return-from-interrupt (RTI) instruction, the CPU can recognize the new interrupt request. **Figure 4-1** shows the logic for external interrupts.

Figure 4-1 shows an external interrupt functional diagram. **Figure 4-2** shows an external interrupt timing diagram for the interrupt line. The timing diagram illustrates two treatments of the interrupt line to the processor.

1. Two single pulses on the interrupt line are spaced far enough apart to be serviced. The minimum time between pulses is a function of the length of the interrupt service.

   Once a pulse occurs, the next pulse normally should not occur until an RTI occurs. This time (t_{ILIL}) is obtained by adding 19 instruction cycles to the total number of cycles needed to complete the service routine (not including the RTI instruction).

2. Many interrupt lines are “wire-ORed” to the $\overline{IRQ}$ line. If the interrupt line remains low after servicing an interrupt, then the CPU continues to recognize an interrupt.

NOTE: *The internal interrupt latch is cleared in the first part of the interrupt service routine. Therefore, a new external interrupt pulse could be latched and serviced as soon as the I bit is cleared.*

If the $\overline{IRQ}$ pin is not in use, connect it to the V_{DD} pin.

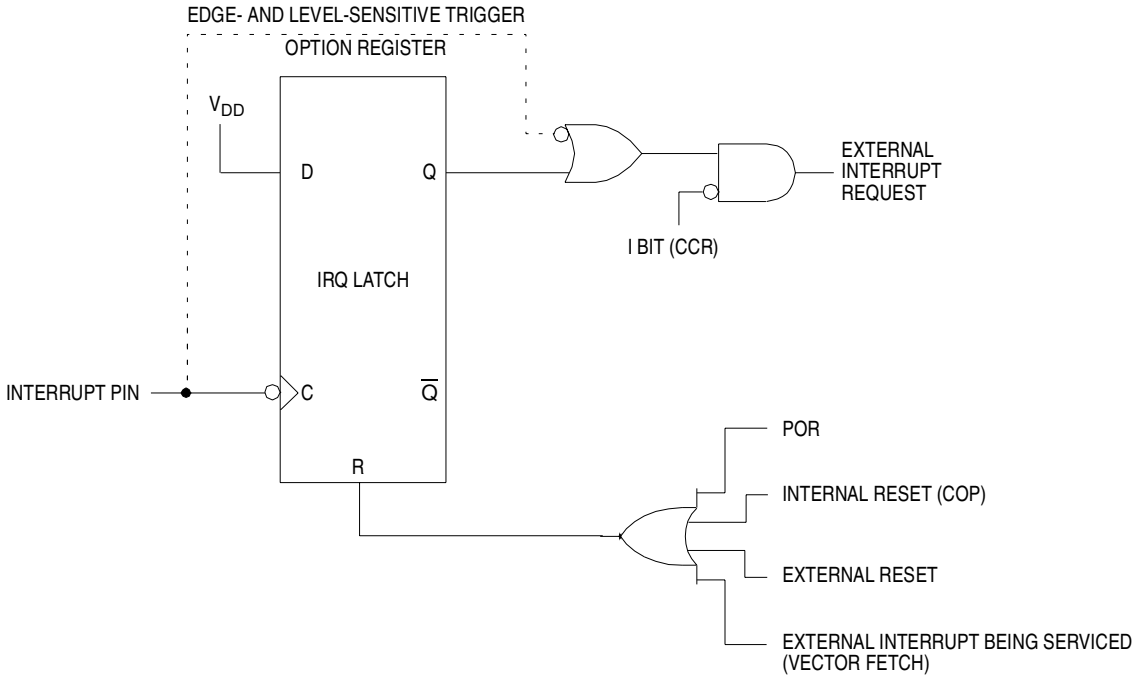
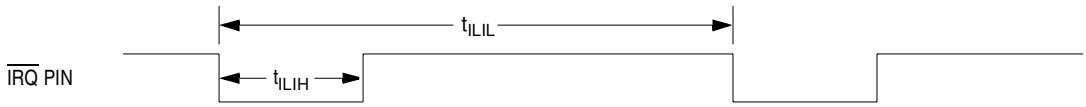
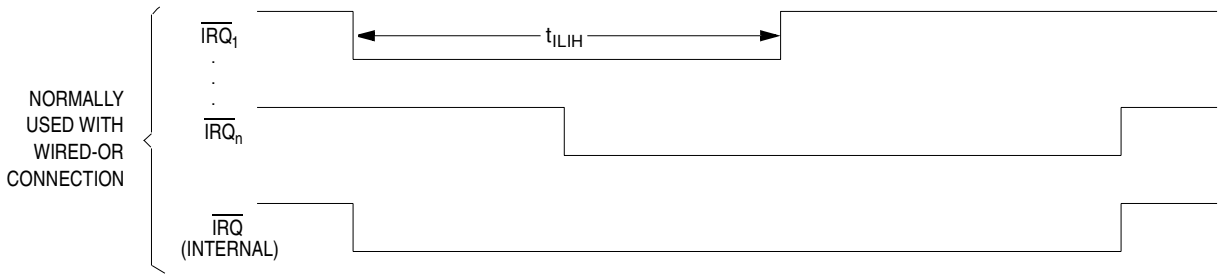


Figure 4-1. External Interrupt Internal Function Diagram



a. Edge-Sensitive Trigger Condition. The minimum pulse width (t_{ILIH}) is either 125 ns (f_{OP} = 2.1 MHz) or 250 ns (f_{OP} = 1 MHz). The period t_{ILIL} should not be less than the number of t_{CYC} cycles it takes to execute the interrupt service routine plus 19 t_{CYC} cycles.



b. Level-Sensitive Trigger Condition. If the interrupt line remains low after servicing an interrupt, then the CPU continues to recognize an interrupt.

Figure 4-2. External Interrupt Timing

4.3.3 Port B Interrupts

When these three conditions are true, a port B pin (PBx) acts as an external interrupt pin:

- The corresponding port B pullup bit (PBPUx) in mask option register 1 (MOR1) is programmed to a logic 1.
- The corresponding port B data direction bit (DDRBx) in data direction register B (DDRB) is a logic 0.
- The clear interrupt mask (CLI) instruction has cleared the I bit in the CCR.

MOR1 is an erasable, programmable read-only memory (EPROM) register that enables the port B pullup device. Data from MOR1 is latched on the rising edge of the voltage on the $\overline{\text{RESET}}$ pin. See **9.5.2 Mask Option Register 1**.

Port B external interrupt pins can be falling-edge sensitive only or both falling-edge and low-level sensitive, depending on the state of the IRQ bit in the option register at location $1FDF.

When the IRQ bit is a logic 1, a falling edge or a low level on a port B external interrupt pin latches an external interrupt request. As long as any port B external interrupt pin is low, an external interrupt request is present, and the CPU continues to execute the interrupt service routine.

When the IRQ bit is a logic 0, a falling-edge only on a port B external interrupt pin latches an external interrupt request. A subsequent port B external interrupt request can be latched only after the voltage level of the previous port B external interrupt signal returns to a logic 1 and then falls again to a logic 0.

Figure 4-3 shows the port B input/output (I/O) logic.

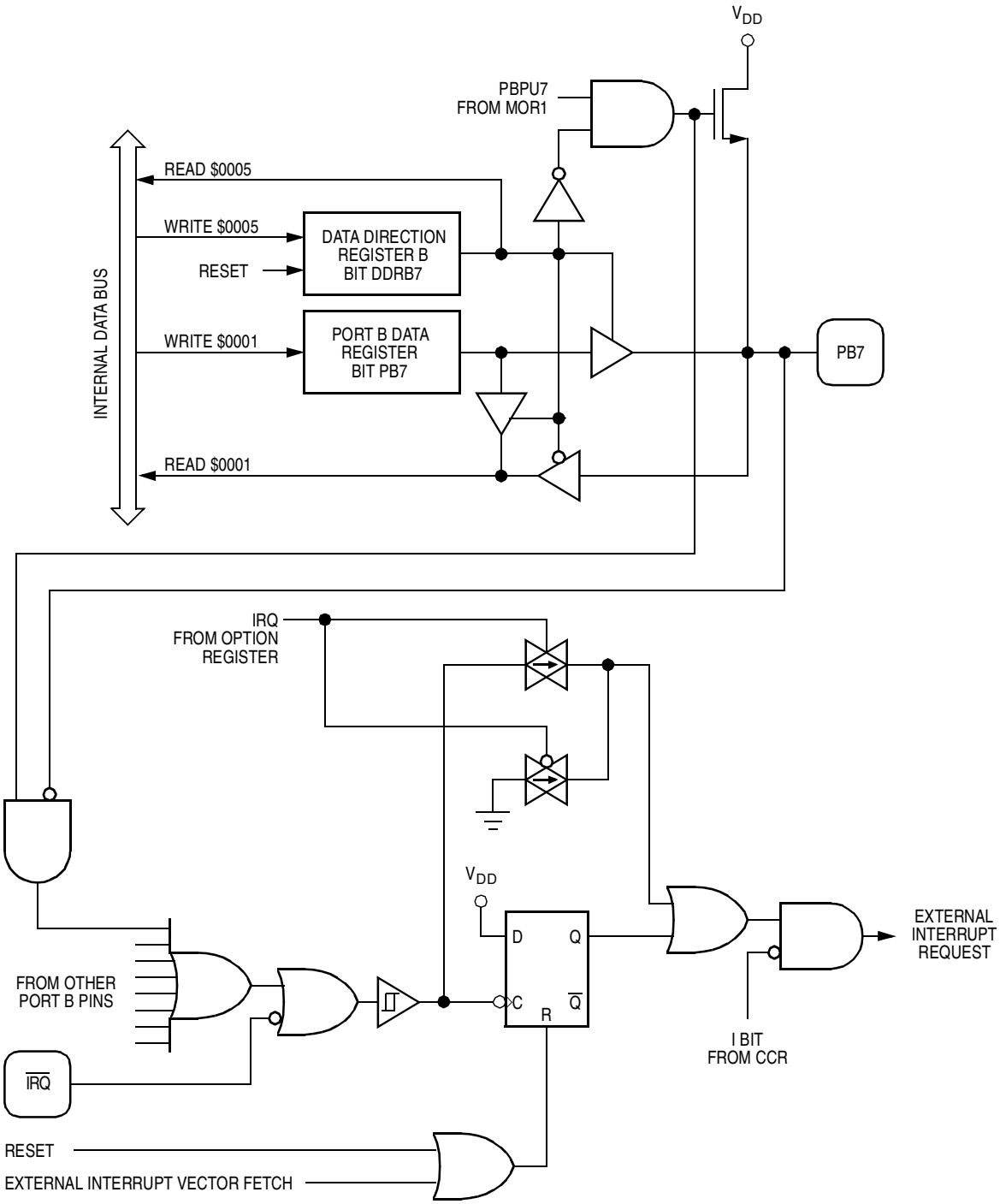


Figure 4-3. Port B I/O Logic

4.3.4 Capture/Compare Timer Interrupts

Setting the I bit in the CCR disables all interrupts except for SWI.

4.3.5 SCI Interrupts

The serial communications interface (SCI) can generate these interrupts:

- Transmit data register empty interrupt
- Transmission complete interrupt
- Receive data register full interrupt
- Receiver overrun interrupt
- Receiver input idle interrupt

Setting the I bit in the CCR disables all SCI interrupts.

- **SCI Transmit Data Register Empty Interrupt** — The transmit data register empty bit (TDRE) indicates that the SCI data register is ready to receive a byte for transmission. TDRE becomes set when data in the SCI data register transfers to the transmit shift register. TDRE generates an interrupt request if the transmit interrupt enable bit (TIE) is set also.
- **SCI Transmission Complete Interrupt** — The transmission complete bit (TC) indicates the completion of an SCI transmission. TC becomes set when the TDRE bit becomes set and no data, preamble, or break character is being transmitted. TC generates an interrupt request if the transmission complete interrupt enable bit (TCIE) is set also.
- **SCI Receive Data Register Full Interrupt** — The receive data register full bit (RDRF) indicates that a byte is ready to be read in the SCI data register. RDRF becomes set when the data in the receive shift register transfers to the SCI data register. RDRF generates an interrupt request if the receive interrupt enable bit (RIE) is set also.

- **SCI Receiver Overrun Interrupt** — The overrun bit (OR) indicates that a received byte is lost because software has not read the previously received byte. OR becomes set when a byte shifts into the receive shift register before software reads the word already in the SCI data register. OR generates an interrupt request if the receive interrupt enable bit (RIE) is set also.
- **SCI Receiver Input Idle Interrupt** — The receiver input idle bit (IDLE) indicates that the SCI receiver input is not receiving data. IDLE becomes set when 10 or 11 consecutive logic 1s appear on the receiver input. IDLE generates an interrupt request if the idle line interrupt enable bit (ILIE) is set also.

4.3.6 SPI Interrupts

The serial peripheral interrupt (SPI) can generate these interrupts:

- SPI transmission complete interrupt
- SPI mode fault interrupt

Setting the I bit in the CCR disables all SPI interrupts.

- **SPI Transmission Complete Interrupt** — The SPI flag bit (SPIF) in the SPI status register indicates the completion of an SPI transmission. SPIF becomes set when a byte shifts into or out of the SPI data register. SPIF generates an interrupt request if the SPIE bit is set also.
- **SPI Mode Fault Interrupt** — The mode fault bit (MODF) in the SPI status register indicates an SPI mode error. MODF becomes set when a logic 0 occurs on the PD5/$\overline{SS}$ pin while the master bit (MSTR) in the SPI control register is set. MODF generates an interrupt request if the SPIE bit is set also.

4.4 Interrupt Processing

The CPU takes these actions to begin servicing an interrupt:

1. Stores the CPU registers on the stack in the order shown in **Figure 4-4**
2. Sets the I bit in the CCR to prevent further interrupts
3. Loads the program counter with the contents of the appropriate interrupt vector locations as shown in **Table 4-1**.

Table 4-1. Reset/Interrupt Vector Addresses

Function	Source	Local Mask	Global Mask	Priority (1 = Highest)	Vector Address
Reset	Power-on logic	None	None	1	$1FFE–$1FFF
	$\overline{\text{RESET}}$ pin				
Software interrupt (SWI)	User code	None	None	Same priority as any instruction	$1FFC–$1FFD
External interrupt	$\overline{\text{IRQ}}$ pin	None	I bit	2	$1FFA–$1FFB
	Port B pins				
Timer interrupts	ICF bit	ICIE bit	I bit	3	$1FF8–$1FF9
	OCF bit	OCIE bit			
	TOF bit	TOIE bit			
SCI interrupts	TDRE bit	TCIE bit	I bit	4	$1FF6–$1FF7
	TC bit				
	RDRF bit	RIE bit			
	OR bit				
	IDLE bit	ILIE bit			
SPI interrupts	SPIF bit	SPIE	I bit	5	$1FF4–$1FF5
	MODF bit				

The return-from-interrupt (RTI) instruction causes the CPU to recover the CPU registers from the stack as shown in **Figure 4-4**.

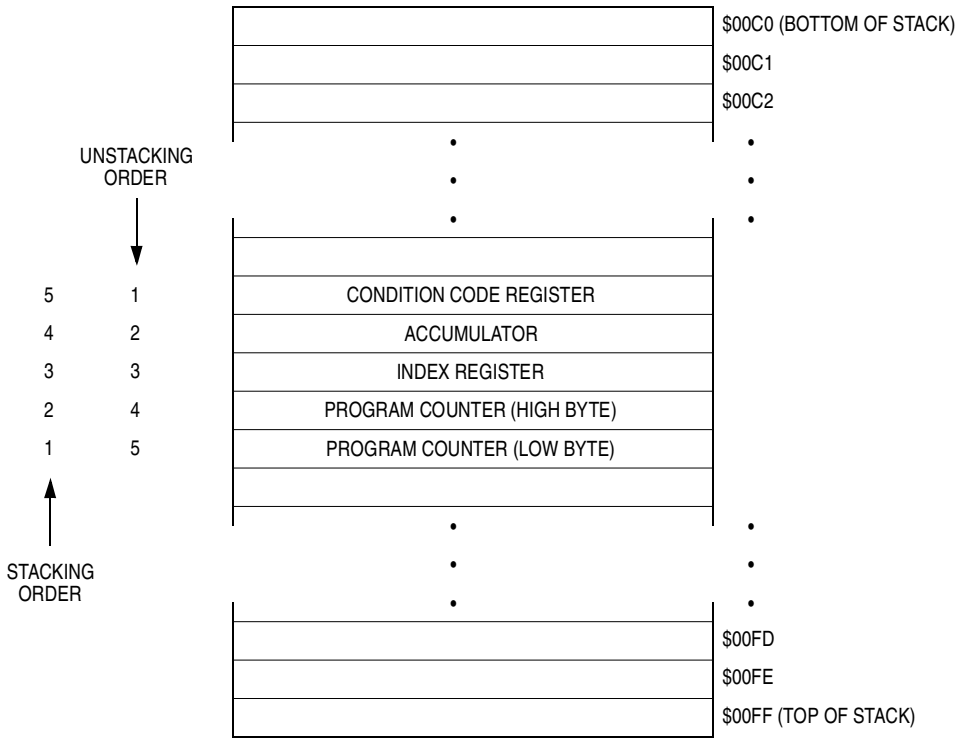


Figure 4-4. Interrupt Stacking Order

NOTE: *If more than one interrupt request is pending, the CPU fetches the vector of the higher priority interrupt first. A higher priority interrupt does not interrupt a lower priority interrupt service routine unless the lower priority interrupt service routine clears the I bit. See* ***Table 4-1*** *for a priority listing.*

Figure 4-5 shows the sequence of events caused by an interrupt.

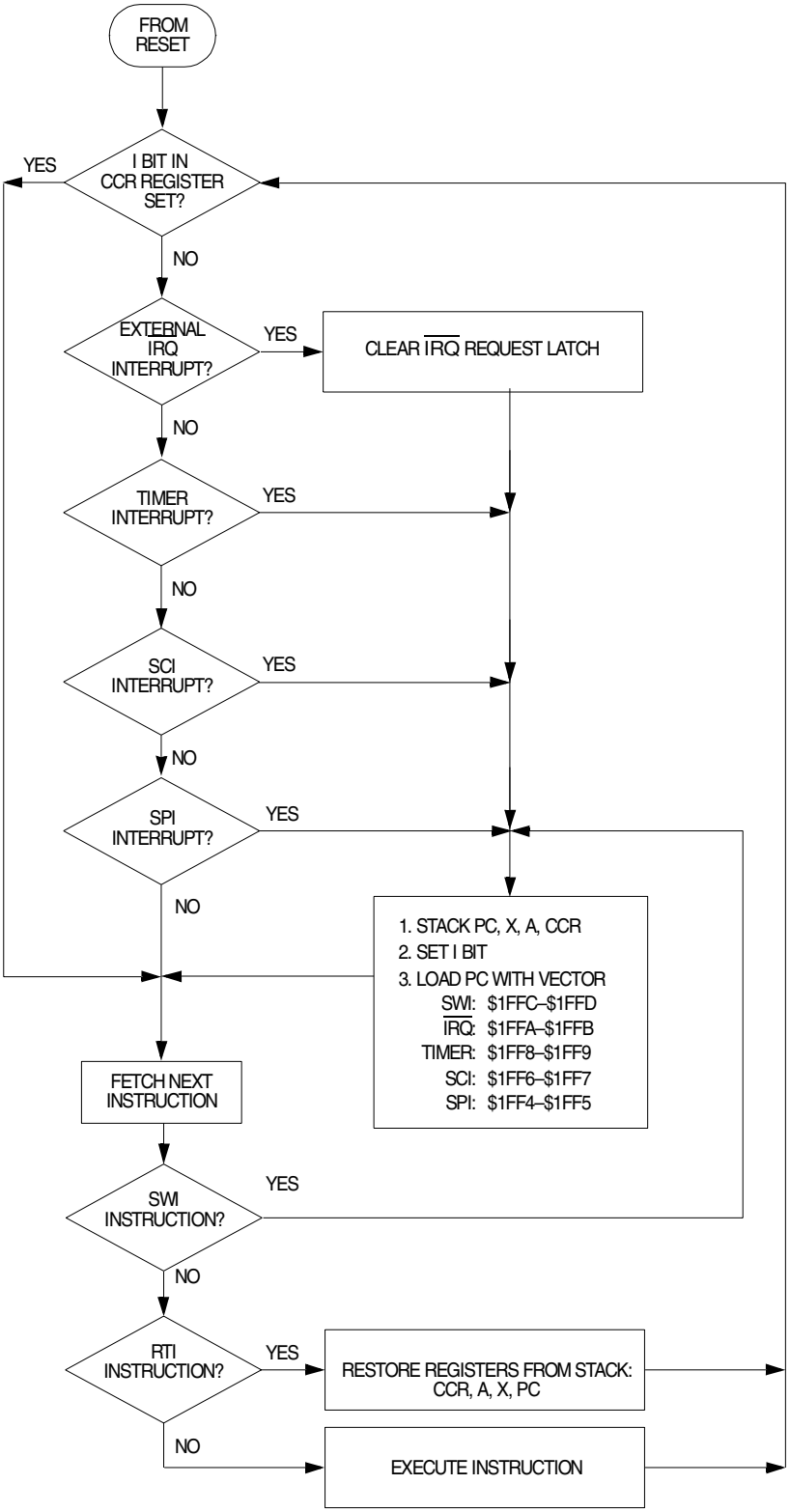


Figure 4-5. Reset and Interrupt Processing Flowchart

Interrupts

Section 5. Resets

5.1 Contents

5.2 Introduction

This section describes how resets initialize the microcontroller unit (MCU).

5.3 Reset Sources

A reset immediately stops the operation of the instruction being executed, initializes certain control bits, and loads the program counter with a user-defined reset vector address. These conditions produce a reset:

- Power-on reset (POR) — Initial power-up
- External reset — A logic 0 applied to the $\overline{\text{RESET}}$ pin
- Internal programmable computer operating properly (COP) watchdog timer reset
- Internal non-programmable COP watchdog timer reset
- Internal clock monitor reset

5.3.1 Power-On Reset (POR)

A positive transition on the V_{DD} pin generates a power-on reset (POR). The POR is strictly for the power-up condition and cannot be used to detect drops in power supply voltage.

A 4064 t_{CYC} (internal clock cycle) delay after the oscillator becomes active allows the clock generator to stabilize. If the $\overline{\text{RESET}}$ pin is at logic 0 at the end of 4064 t_{CYC}, the MCU remains in the reset condition until the signal on the $\overline{\text{RESET}}$ pin goes to logic 1.

5.3.2 External Reset

The minimum time required for the MCU to recognize a reset is 1 1/2 t_{CYC}. However, to guarantee that the MCU recognizes an external reset as an external reset and not as a COP or clock monitor reset, the $\overline{\text{RESET}}$ pin must be low for eight t_{CYC}. After six t_{CYC}, the input on the $\overline{\text{RESET}}$ pin is sampled. If the pin is still low, an external reset has occurred. If the input is high, then the MCU assumes that the reset was initiated internally by either the COP watchdog timer or by the clock monitor. This method of differentiating between external and internal reset conditions assumes that the $\overline{\text{RESET}}$ pin will rise to a logic 1 less than two t_{CYC} after its release and that an externally generated reset should stay active for at least eight t_{CYC}.

5.3.3 Programmable and Non-Programmable COP Watchdog Resets

A timeout of a COP watchdog generates a COP reset. A COP watchdog, once enabled, is part of a software error detection system and must be cleared periodically to start a new timeout period.

The MC68HC705C8A has two different COP watchdogs for compatibility with devices such as the MC68HC705C8 and the MC68HC05C4A:

1. Programmable COP watchdog reset
2. Non-programmable COP watchdog

One COP has four programmable timeout periods and the other has a fixed non-programmable timeout period.

5.3.3.1 Programmable COP Watchdog Reset

A timeout of the 18-stage ripple counter in the programmable COP watchdog generates a reset. **Figure 5-1** is a diagram of the programmable COP watchdog. Two registers control and monitor operation of the programmable COP watchdog:

- COP reset register (COPRST), $001D
- COP control register (COPCR), $001E

To clear the programmable COP watchdog and begin a new timeout period, write these values to the COP reset register (COPRST). See **Figure 5-2**.

1. $55
2. $AA

The $55 write must precede the $AA write. Instructions may be executed between the write operations provided that the COP watchdog does not time out before the second write.

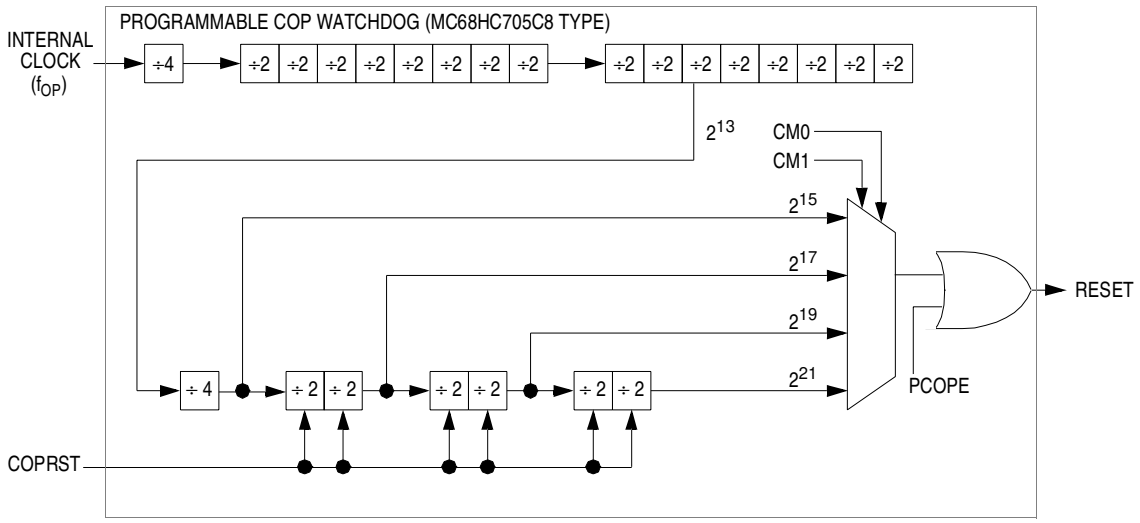


Figure 5-1. Programmable COP Watchdog Diagram

Address: $001D

	Bit 7	6	5	4	3	2	1	Bit 0
Read:								
Write:	Bit 7	6	5	4	3	2	1	Bit 0
Reset:	U	U	U	U	U	U	U	U

= Unimplemented U = Unaffected

Figure 5-2. Programmable COP Reset Register (COPRST)

The programmable COP control register (COPCR) shown in **Figure 5-3** does these functions:

- Flags programmable COP watchdog resets
- Enables the clock monitor
- Enables the programmable COP watchdog
- Controls the timeout period of the programmable COP watchdog

Address: $001E

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	COPF	CME	PCOPE	CM1	CM0
Write:								
Reset:	0	0	0	U	0	0	0	0

= Unimplemented U = Unaffected

Figure 5-3. Programmable COP Control Register (COPCR)

COPF — COP Flag

This read-only bit is set when a timeout of the programmable COP watchdog occurs or when the clock monitor detects a slow or absent internal clock. Clear the COPF bit by reading the COP control register. Reset has no effect on the COPF bit.

1 = COP timeout or internal clock failure

0 = No COP timeout and no internal clock failure

CME — Clock Monitor Enable Bit

This read/write bit enables the clock monitor. The clock monitor sets the COPF bit and generates a reset if it detects an absent internal clock for a period of from 5 μs to 100 μs. CME is readable and writable at any time. Reset clears the CME bit.

1 = Clock monitor enabled
0 = Clock monitor disabled

NOTE: *Do not enable the clock monitor in applications with an internal clock frequency of 200 kHz or less.*

If the clock monitor detects a slow clock, it drives the bidirectional $\overline{\text{RESET}}$ pin low for four clock cycles. If the clock monitor detects an absent clock, it drives the $\overline{\text{RESET}}$ pin low until the clock recovers.

PCOPE — Programmable COP Enable Bit

This read/write bit enables the programmable COP watchdog. PCOPE is readable at any time but can be written only once after reset. Reset clears the PCOPE bit.

1 = Programmable COP watchdog enabled
0 = Programmable COP watchdog disabled

NOTE: *Programming the non-programmable COP enable bit (NCOPE) in mask option register 2 (MOR2) to logic 1 enables the non-programmable COP watchdog. Setting the PCOPE bit while the NCOPE bit is programmed to logic 1 enables both COP watchdogs to operate at the same time. (See **9.5.3 Mask Option Register 2**.)*

CM1 and CM0 — COP Mode Bits

These read/write bits select the timeout period of the programmable COP watchdog. (See **Table 5-1**.) CM1 and CM0 can be read anytime but can be written only once. They can be cleared only by reset.

Bits 7–5 — Unused

Bits 7–5 always read as logic 0s. Reset clears bits 7–5.

Table 5-1. Programmable COP Timeout Period Selection

CM1:CM0	COP Timeout Rate	Programmable COP Timeout Period			
		f_{OSC} = 4.0 MHz f_{OP} = 2.0 MHz	f_{OSC} = 3.5795 MHz f_{OP} = 1.7897 MHz	f_{OSC} = 2.0 MHz f_{OP} = 1.0 MHz	f_{OSC} = 1.0 MHz f_{OP} = 0.5 MHz
00	$f_{OP} \div 2^{15}$	16.38 ms	18.31 ms	32.77 ms	65.54 ms
01	$f_{OP} \div 2^{17}$	65.54 ms	73.24 ms	131.07 ms	262.14 ms
10	$f_{OP} \div 2^{19}$	262.14 ms	292.95 ms	524.29 ms	1.048 s
11	$f_{OP} \div 2^{21}$	1.048 s	1.172 s	2.097 s	4.194 s

5.3.3.2 Non-Programmable COP Watchdog

A timeout of the 18-stage ripple counter in the non-programmable COP watchdog generates a reset. The timeout period is 65.536 ms when f_{OSC} = 4 MHz. The timeout period for the non-programmable COP timer is a direct function of the crystal frequency. The equation is:

$$\text{Timeout period} = \frac{262{,}144}{f_{OSC}}$$

Two memory locations control operation of the non-programmable COP watchdog:

1. Non-programmable COP enable bit (NCOPE) in mask option register 2 (MOR2)

   Programming the NCOPE bit in MOR2 to a logic 1 enables the non-programmable COP watchdog. See **9.5.3 Mask Option Register 2**.

NOTE: *Writing a logic 1 to the programmable COP enable bit (PCOPE) in the COP control register enables the programmable COP watchdog. Setting the PCOPE bit while the NCOPE bit is programmed to logic 1 enables both COP watchdogs to operate at the same time.*

2. COP clear bit (COPC) at address $1FF0

   To clear the non-programmable COP watchdog and start a new COP timeout period, write a logic 0 to bit 0 of address $1FF0. Reading address $1FF0 returns the mask option register 1 (MOR1) data at that location. See **9.5.2 Mask Option Register 1**.

NOTE: *The non-programmable watchdog COP is disabled in bootloader mode, even if the NCOPE bit is programmed.*

Figure 5-4 is a diagram of the non-programmable COP.

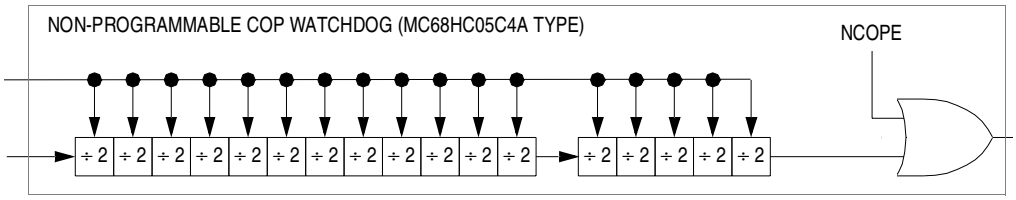


Figure 5-4. Non-Programmable COP Watchdog Diagram

5.3.4 Clock Monitor Reset

When the CME bit in the COP control register is set, the clock monitor detects the absence of the internal bus clock for a certain period of time. The timeout period depends on processing parameters and varies from 5 µs to 100 µs, which implies that systems using a bus clock rate of 200 kHz or less should not use the clock monitor function.

If a slow or absent clock is detected, the clock monitor causes a system reset. The reset is issued to the external system for four bus cycles using the bidirectional $\overline{\text{RESET}}$ pin.

Special consideration is required when using the STOP instruction with the clock monitor. Since STOP causes the system clocks to halt, the clock monitor issues a system reset when STOP is executed.

The clock monitor is a useful backup to the COP watchdog system. Because the watchdog timer requires a clock to function, it cannot indicate a system clock failure. The clock monitor would detect such a condition and force the MCU to a reset state. Clocks are not required for the MCU to reach a reset condition. They are, however, required to bring the MCU through the reset sequence and back to run condition.

Section 6. Low-Power Modes

6.1 Contents

6.2 Introduction

This section describes the three low-power modes:

- Stop mode
- Wait mode
- Data-retention mode

6.3 Stop Mode

The STOP instruction places the microcontroller unit (MCU) in its lowest power consumption mode. In stop mode, the internal oscillator is turned off, halting all internal processing including timer, serial communications interface (SCI), and master mode serial peripheral interface (SPI) operation. See **Figure 6-1**.

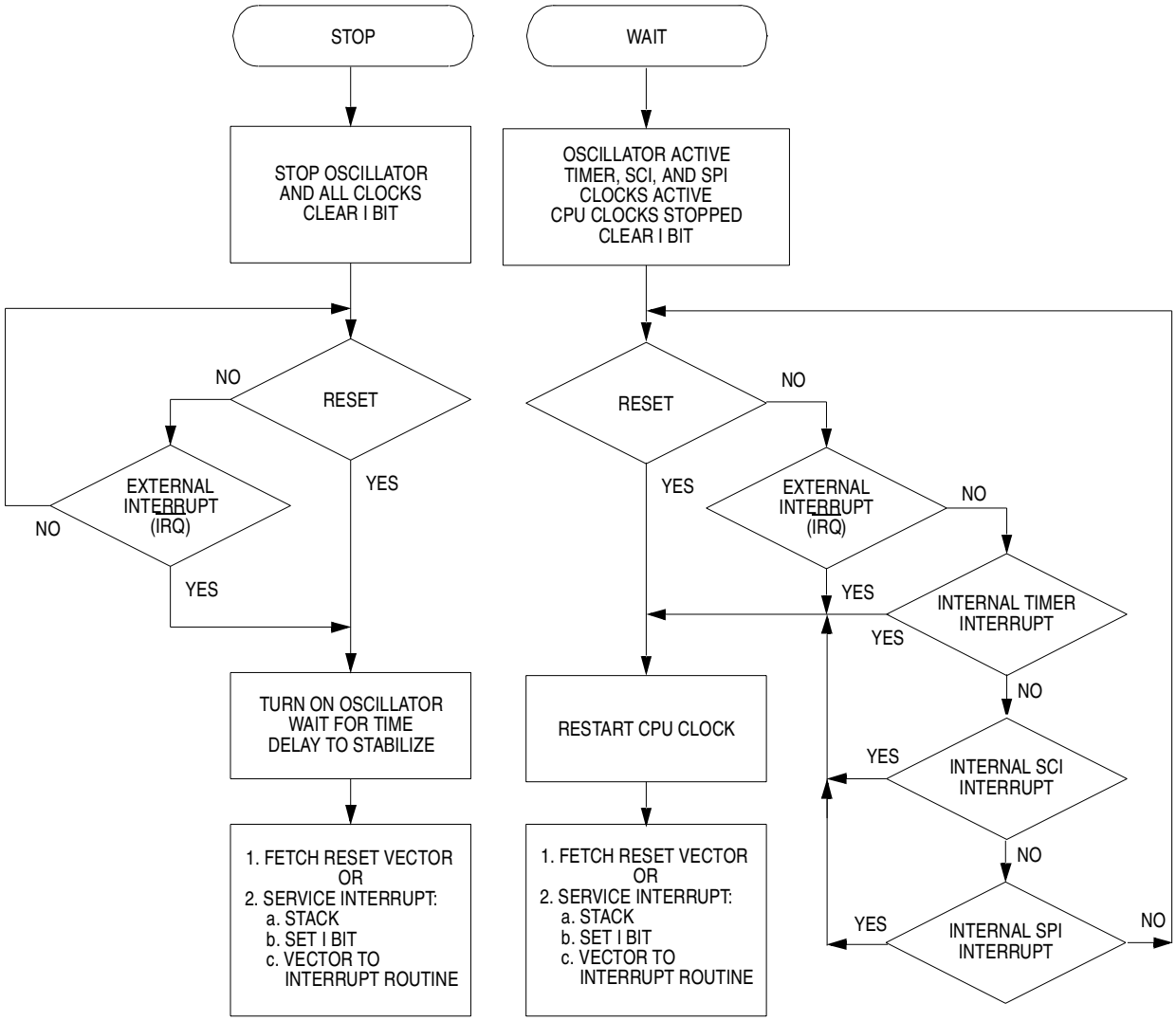


Figure 6-1. Stop/Wait Mode Function Flowchart

During stop mode, the I bit in the condition code register (CCR) is cleared to enable external interrupts. All other registers and memory remain unaltered. All input/output (I/O) lines remain unchanged. The processor can be brought out of stop mode only by an external interrupt or reset.

6.3.1 SCI During Stop Mode

When the MCU enters stop mode, the baud rate generator stops, halting all SCI activity. If the STOP instruction is executed during a transmitter transfer, that transfer is halted. If a low input to the $\overline{\text{IRQ}}$ pin is used to exit stop mode, the transfer resumes.

If the SCI receiver is receiving data and stop mode is entered, received data sampling stops because the baud rate generator stops, and all subsequent data is lost. Therefore, all SCI transfers should be in the idle state when the STOP instruction is executed.

6.3.2 SPI During Stop Mode

When the MCU enters stop mode, the baud rate generator stops, terminating all master mode SPI operations. If the STOP instruction is executed during an SPI transfer, that transfer halts until the MCU exits stop mode by a low signal on the $\overline{\text{IRQ}}$ pin. If reset is used to exit stop mode, the SPI control and status bits are cleared, and the SPI is disabled.

If the MCU is in slave mode when the STOP instruction is executed, the slave SPI continues to operate and can still accept data and clock information in addition to transmitting its own data back to a master device. At the end of a possible transmission with a slave SPI in stop mode, no flags are set until a low on the $\overline{\text{IRQ}}$ pin wakes up the MCU.

NOTE: *Although a slave SPI in stop mode can exchange data with a master SPI, the status bits of a slave SPI are inactive in stop mode.*

6.3.3 Programmable COP Watchdog in Stop Mode

The STOP instruction turns off the internal oscillator and suspends the computer operating properly (COP) watchdog counter. If the $\overline{\text{RESET}}$ pin brings the MCU out of stop mode, the reset function clears and disables the COP watchdog.

If the $\overline{\text{IRQ}}$ pin brings the MCU out of stop mode, the COP counter resumes counting from its suspended value after the 4064-t_{CYC} clock stabilization delay. See **Figure 6-2**.

NOTE: *If the clock monitor is enabled (CME = 1), the STOP instruction causes the clock monitor to time out and reset the MCU.*

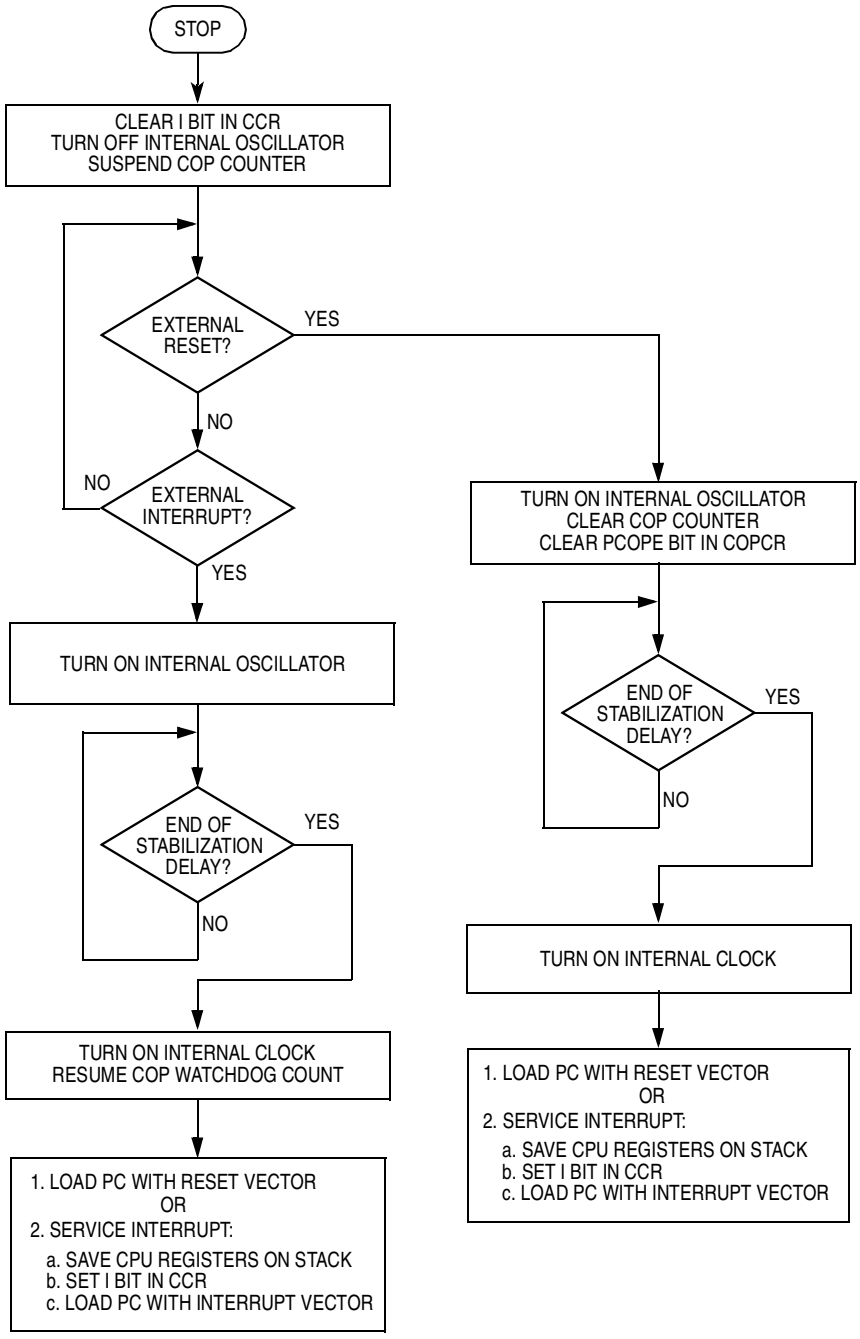


Figure 6-2. Programmable COP Watchdog in Stop Mode (PCOPE = 1) Flowchart

6.3.4 Non-Programmable COP Watchdog in Stop Mode

The STOP instruction has these effects on the non-programmable COP watchdog:

- Turns off the oscillator and the COP watchdog counter
- Clears the COP watchdog counter

If the $\overline{\text{RESET}}$ pin brings the MCU out of stop mode, the COP watchdog begins counting immediately. The reset function clears the COP counter again after the 4064-t_{CYC} clock stabilization delay.

If the $\overline{\text{IRQ}}$ pin brings the MCU out of stop mode, the COP watchdog begins counting immediately. The IRQ function does not clear the COP counter again after the 4064-t_{CYC} clock stabilization delay. See **Figure 6-3**.

NOTE: *If the clock monitor is enabled (CME = 1), the STOP instruction causes it to time out and reset the MCU.*

6.4 Wait Mode

The WAIT instruction places the MCU in an intermediate power consumption mode. All central processor unit (CPU) activity is suspended, but the oscillator, capture/compare timer, SCI, and SPI remain active. Any interrupt or reset brings the MCU out of wait mode. See **Figure 6-1**.

The WAIT instruction has these effects on the CPU:

- Clears the I bit in the condition code register, enabling interrupts
- Stops the CPU clock, but allows the internal clock to drive the capture/compare timer, SCI, and SPI

The WAIT instruction does not affect any other registers or I/O lines. The capture/compare timer, SCI, and SPI can be enabled to allow a periodic exit from wait mode.

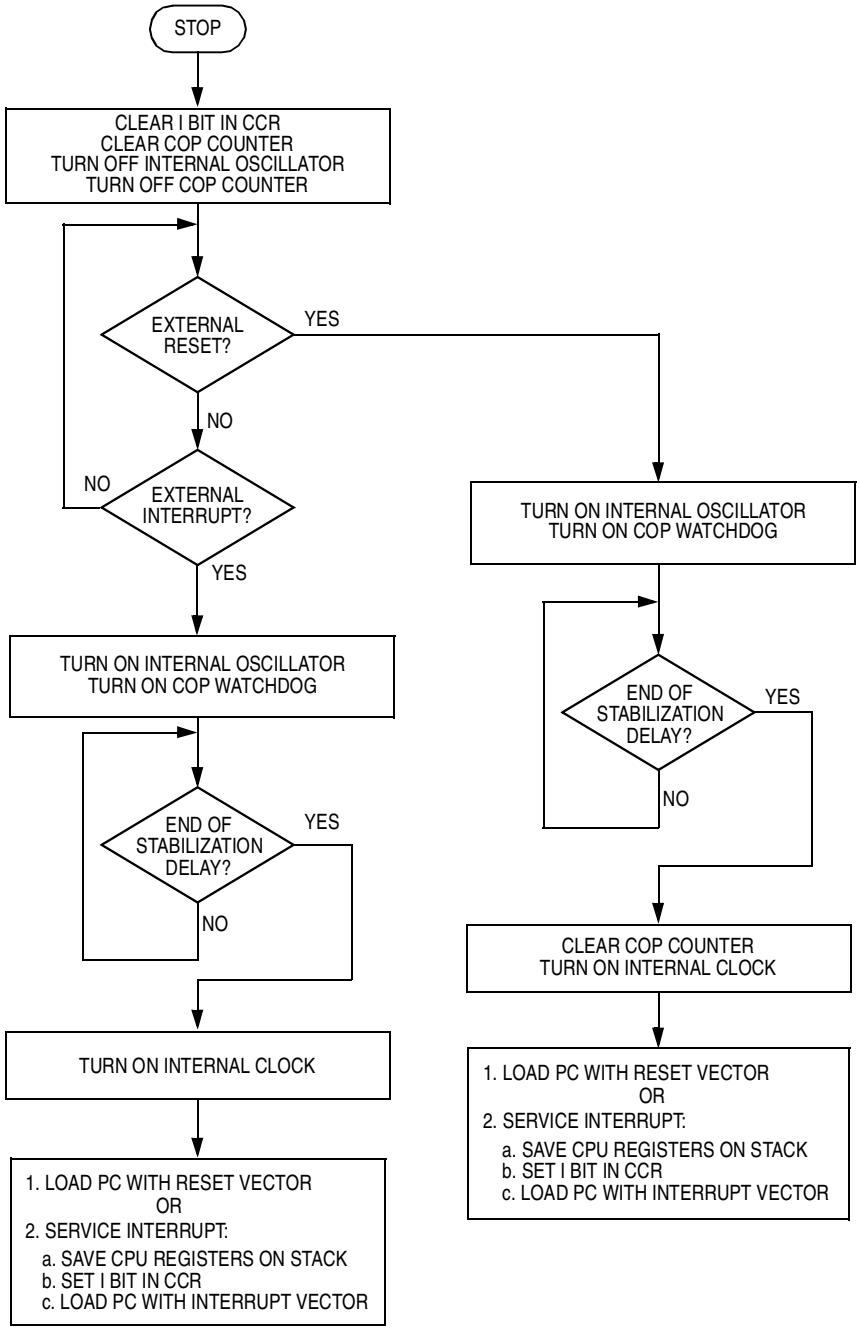


Figure 6-3. Non-Programmable COP Watchdog in Stop Mode (NCOPE = 1) Flowchart

6.4.1 Programmable COP Watchdog in Wait Mode

The programmable COP watchdog is active during wait mode. Software must periodically bring the MCU out of wait mode to clear the programmable COP watchdog.

6.4.2 Non-Programmable COP Watchdog in Wait Mode

The non-programmable COP watchdog is active during wait mode. Software must periodically bring the MCU out of wait mode to clear the non-programmable COP watchdog.

6.5 Data-Retention Mode

In data-retention mode, the MCU retains random-access memory (RAM) contents and CPU register contents at V_{DD} voltages as low as 2.0 Vdc. The data-retention feature allows the MCU to remain in a low power-consumption state during which it retains data, but the CPU cannot execute instructions.

To put the MCU in data-retention mode:

1. Drive the $\overline{\text{RESET}}$ pin to logic 0.
2. Lower V_{DD} voltage. The $\overline{\text{RESET}}$ pin must remain low continuously during data-retention mode.

To take the MCU out of data-retention mode:

1. Return V_{DD} to normal operating voltage.
2. Return the $\overline{\text{RESET}}$ pin to logic 1.

Section 7. Parallel Input/Output (I/O)

7.1 Contents

7.2 Introduction

This section describes the programming of ports A, B, C, and D.

7.3 Port A

Port A is an 8-bit, general-purpose, bidirectional input/output (I/O) port.

7.3.1 Port A Data Register

The port A data register (PORTA) shown in **Figure 7-1** contains a data latch for each of the eight port A pins. When a port A pin is programmed to be an output, the state of its data register bit determines the state of the output pin. When a port A pin is programmed to be an input, reading the port A data register returns the logic state of the pin.

Address: $0000

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
Reset:	Unaffected by reset							

Figure 7-1. Port A Data Register (PORTA)

PA7–PA0 — Port A Data Bits

These read/write bits are software programmable. Data direction of each bit is under the control of the corresponding bit in data direction register A. Reset has no effect on port A data.

7.3.2 Data Direction Register A

The contents of data direction register A (DDRA) shown in **Figure 7-2** determine whether each port A pin is an input or an output. Writing a logic 1 to a DDRA bit enables the output buffer for the associated port A pin; a logic 0 disables the output buffer. A reset clears all DDRA bits, configuring all port A pins as inputs.

Address: $0004

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
Reset:	0	0	0	0	0	0	0	0

Figure 7-2. Data Direction Register A (DDRA)

DDRA7–DDRA0 — Port A Data Direction Bits

These read/write bits control port A data direction. Reset clears bits DDRA7–DDRA0.

1 = Corresponding port A pin configured as output
0 = Corresponding port A pin configured as input

NOTE: *Avoid glitches on port A pins by writing to the port A data register before changing DDRA bits from logic 0 to logic 1.*

7.3.3 Port A Logic

Figure 7-3 is a diagram of the port A I/O logic.

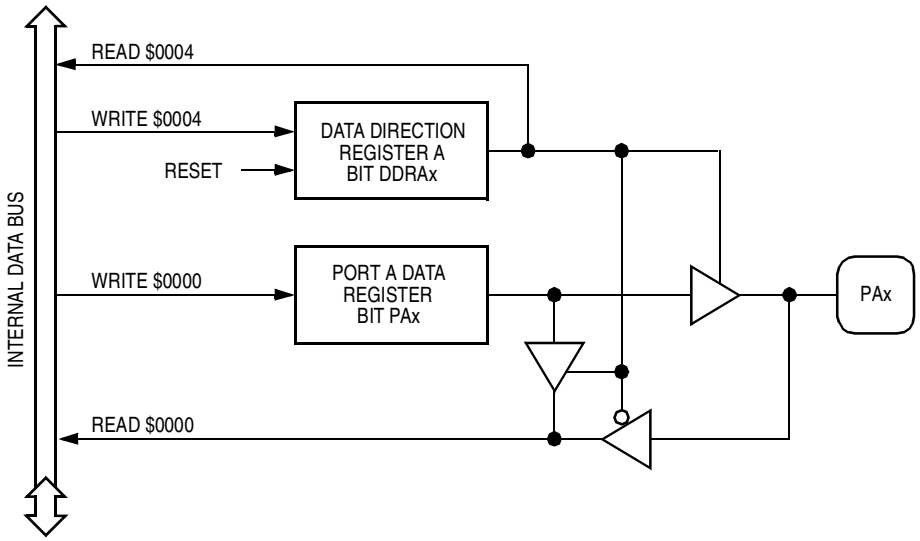


Figure 7-3. Port A I/O Logic

When a port A pin is programmed to be an output, the state of its data register bit determines the state of the output pin. When a port A pin is programmed to be an input, reading the port A data register returns the logic state of the pin.

The data latch can always be written, regardless of the state of its DDRA bit. Table 7-1 summarizes the operation of the port A pins.

Table 7-1. Port A Pin Functions

DDRA Bit	I/O Pin Mode	Accesses to DDRA	Accesses to PORTA	
		Read/Write	Read	Write
0	Input, Hi-Z(1)	DDRA7–DDRA0	Pin	PA7–PA0(2)
1	Output	DDRA7–DDRA0	PA7–PA0	PA7–PA0

1. Hi-Z = high impedance
2. Writing affects data register but does not affect input.

NOTE: *To avoid excessive current draw, tie all unused input pins to V_{DD} or V_{SS}, or change I/O pins to outputs by writing to DDRA in user code as early as possible.*

7.4 Port B

Port B is an 8-bit, general-purpose, bidirectional I/O port. Port B pins can also be configured to function as external interrupts. The port B pullup devices are enabled in mask option register 1 (MOR1). See **9.5.2 Mask Option Register 1** and **4.3.3 Port B Interrupts**.

7.4.1 Port B Data Register

The port B data register (PORTB) shown in **Figure 7-4** contains a data latch for each of the eight port B pins.

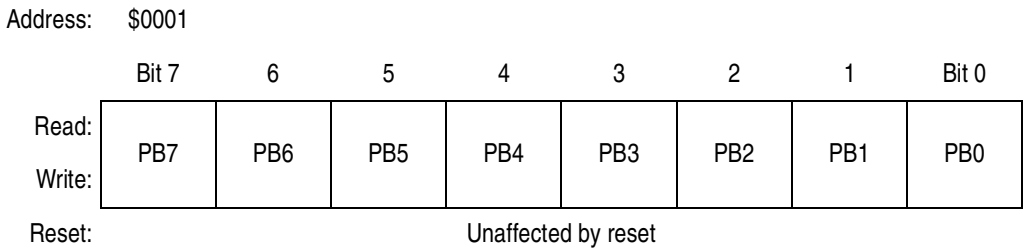


Figure 7-4. Port B Data Register (PORTB)

PB7–PB0 — Port B Data Bits

These read/write bits are software programmable. Data direction of each bit is under the control of the corresponding bit in data direction register B. Reset has no effect on port B data.

7.4.2 Data Direction Register B

The contents of data direction register B (DDRB) shown in **Figure 7-5** determine whether each port B pin is an input or an output. Writing a logic 1 to a DDRB bit enables the output buffer for the associated port B pin; a logic 0 disables the output buffer. A reset clears all DDRB bits, configuring all port B pins as inputs. If the pullup devices are enabled by mask option, setting a DDRB bit to a logic 1 turns off the pullup device for that pin.

Address: $0005

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
Reset:	0	0	0	0	0	0	0	0

Figure 7-5. Data Direction Register B (DDRB)

DDRB7–DDRB0 — Port B Data Direction Bits

These read/write bits control port B data direction. Reset clears bits DDRB7–DDRB0.

1 = Corresponding port B pin configured as output
0 = Corresponding port B pin configured as input

NOTE: *Avoid glitches on port B pins by writing to the port B data register before changing DDRB bits from logic 0 to logic 1.*

7.4.3 Port B Logic

Figure 7-6 shows the port B I/O logic.

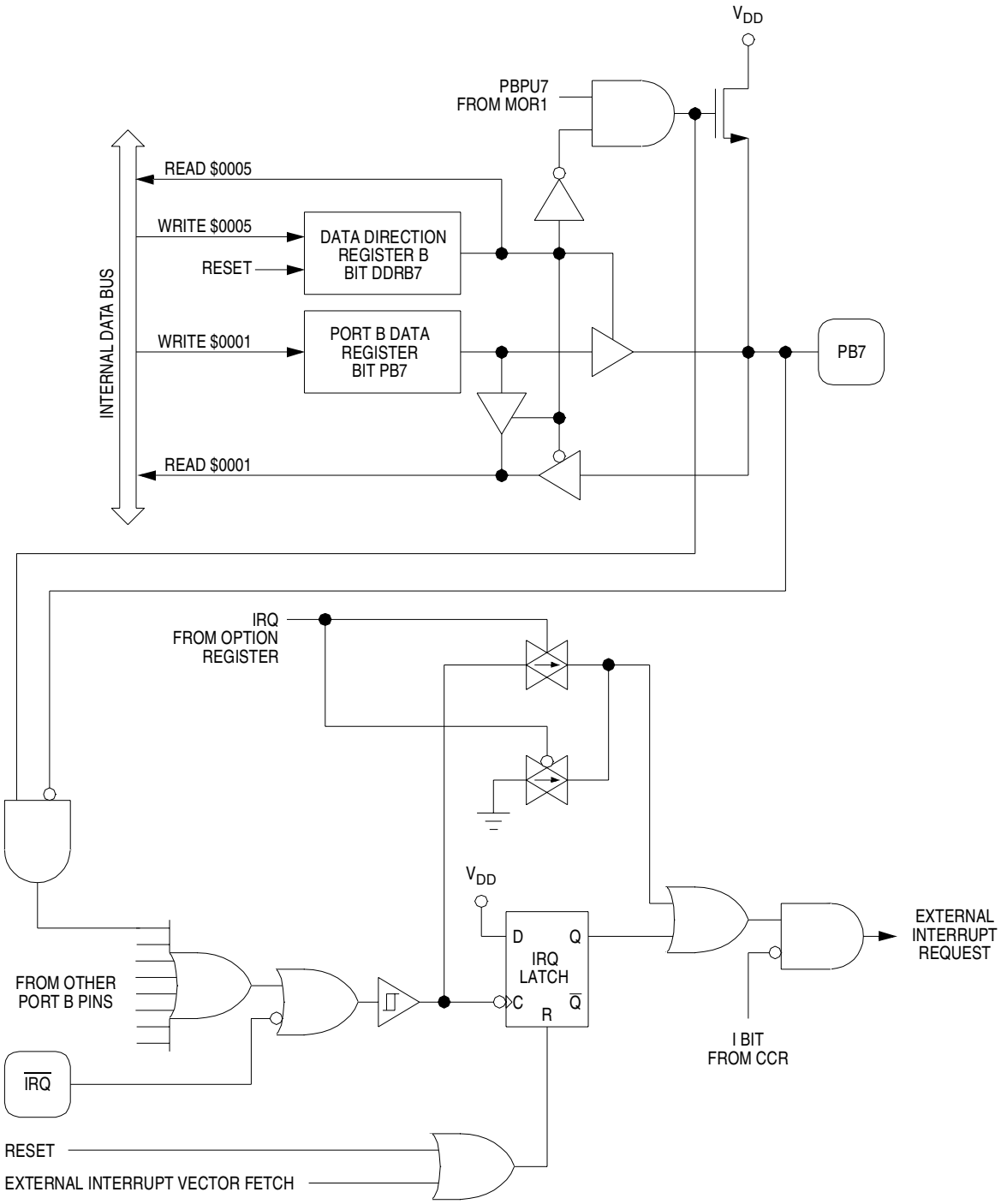


Figure 7-6. Port B I/O Logic

When a port B pin is programmed as an output, reading the port bit reads the value of the data latch and not the voltage on the pin itself. When a port B pin is programmed as an input, reading the port bit reads the voltage level on the pin. The data latch can always be written, regardless of the state of its DDRB bit.

Table 7-2. Port B Pin Functions

DDRB Bit	I/O Pin Mode	Accesses to DDRB	Accesses to PORTB	
		Read/Write	Read	Write
0	Input, Hi-Z(1)	DDRB7–DDRB0	Pin	PB7–PB0(2)
1	Output	DDRB7–DDRB0	PB7–PB0	PB7–PB0

1. Hi-Z = high impedance
2. Writing affects data register but does not affect input.

NOTE: *To avoid excessive current draw, tie all unused input pins to V_{DD} or V_{SS}, or for I/O pins change to outputs by writing to DDRB in user code as early as possible.*

7.5 Port C

Port C is an 8-bit, general-purpose, bidirectional I/O port. PC7 has a high current sink and source capability.

7.5.1 Port C Data Register

The port C data register (PORTC) shown in **Figure 7-7** contains a data latch for each of the eight port C pins. When a port C pin is programmed to be an output, the state of its data register bit determines the state of the output pin. When a port C pin is programmed to be an input, reading the port C data register returns the logic state of the pin.

Address: $0002

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
Reset:	Unaffected by reset							

Figure 7-7. Port C Data Register (PORTC)

PC7–PC0 — Port C Data Bits

These read/write bits are software programmable. Data direction of each bit is under the control of the corresponding bit in data direction register C. PC7 has a high current sink and source capability. Reset has no effect on port C data.

7.5.2 Data Direction Register C

The contents of data direction register C (DDRC) shown in **Figure 7-8** determine whether each port C pin is an input or an output. Writing a logic 1 to a DDRC bit enables the output buffer for the associated port C pin; a logic 0 disables the output buffer. A reset clears all DDRC bits, configuring all port C pins as inputs.

Address: $0006

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	DDRC7	DDRC6	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0
Reset:	0	0	0	0	0	0	0	0

Figure 7-8. Data Direction Register C (DDRC)

DDRC7–DDRC0 — Port C Data Direction Bits

These read/write bits control port C data direction. Reset clears bits DDRC7–DDRC0.

1 = Corresponding port C pin configured as output
0 = Corresponding port C pin configured as input

NOTE: *Avoid glitches on port C pins by writing to the port C data register before changing DDRC bits from logic 0 to logic 1.*

7.5.3 Port C Logic

Figure 7-9 shows port C I/O logic.

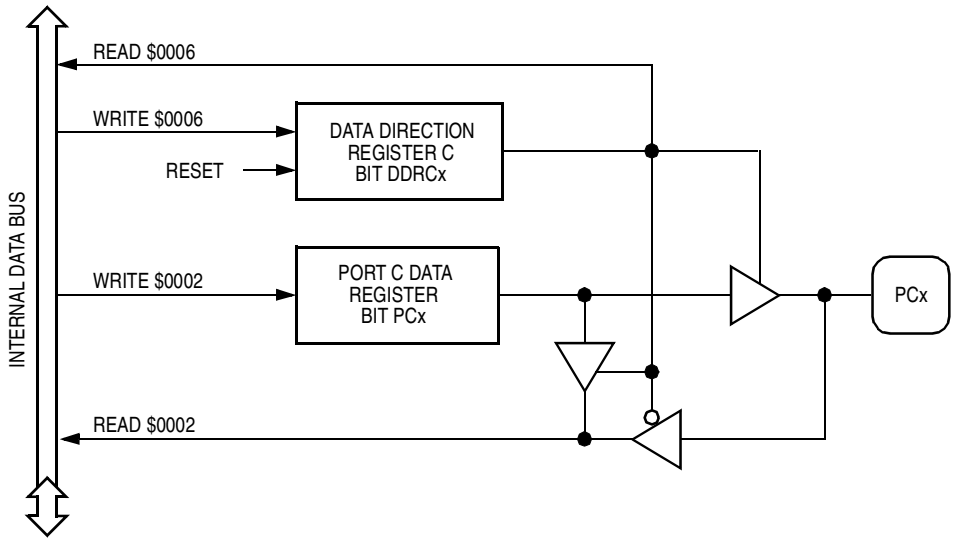


Figure 7-9. Port C I/O Logic

When a port C pin is programmed as an output, reading the port bit reads the value of the data latch and not the voltage on the pin. When a port C pin is programmed as an input, reading the port bit reads the voltage level on the pin. The data latch can always be written, regardless of the state of its DDRC bit. **Table 7-3** summarizes the operation of the port C pins.

Table 7-3. Port C Pin Functions

DDRC Bit	I/O Pin Mode	Accesses to DDRC	Accesses to PORTC	
		Read/Write	Read	Write
0	Input, Hi-Z(1)	DDRC7–DDRC0	Pin	PC7–PC0(2)
1	Output	DDRC7–DDRC0	PC7–PC0	PC7–PC0

1. Hi-Z = high impedance
2. Writing affects data register but does not affect input.

NOTE: *To avoid excessive current draw, tie all unused input pins to V_{DD} or V_{SS} or change I/O pins to outputs by writing to DDRC in user code as early as possible.*

7.6 Port D

Port D is a 7-bit, special-purpose, input-only port that has no data register. Reading address $0003 returns the logic states of the port D pins.

Port D shares pins PD5–PD2 with the serial peripheral interface module (SPI). When the SPI is enabled, PD5–PD2 read as logic 0s. When the SPI is disabled, reading address $0003 returns the logic states of the PD5–PD2 pins.

Port D shares pins PD1 and PD0 with the SCI module. When the SCI is enabled, PD1 and PD0 read as logic 0s. When the SCI is disabled, reading address $0003 returns the logic states of the PD1 and PD0 pins.

Address: $0003

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	PD7		SS	SCK	MOSI	MISO	TDO	RDI
Write:								
Reset:	Unaffected by reset							

= Unimplemented

Figure 7-10. Port D Fixed Input Register (PORTD)

Section 8. Capture/Compare Timer

8.1 Contents

8.2 Introduction

This section describes the operation of the 16-bit capture/compare timer. **Figure 8-1** shows the structure of the timer module. **Figure 8-2** is a summary of the timer input/output (I/O) registers.

8.3 Timer Operation

The core of the capture/compare timer is a 16-bit free-running counter. The counter is the timing reference for the input capture and output compare functions. The input capture and output compare functions can latch the times at which external events occur, measure input waveforms, and generate output waveforms and timing delays. Software can read the value in the counter at any time without affecting the counter sequence.

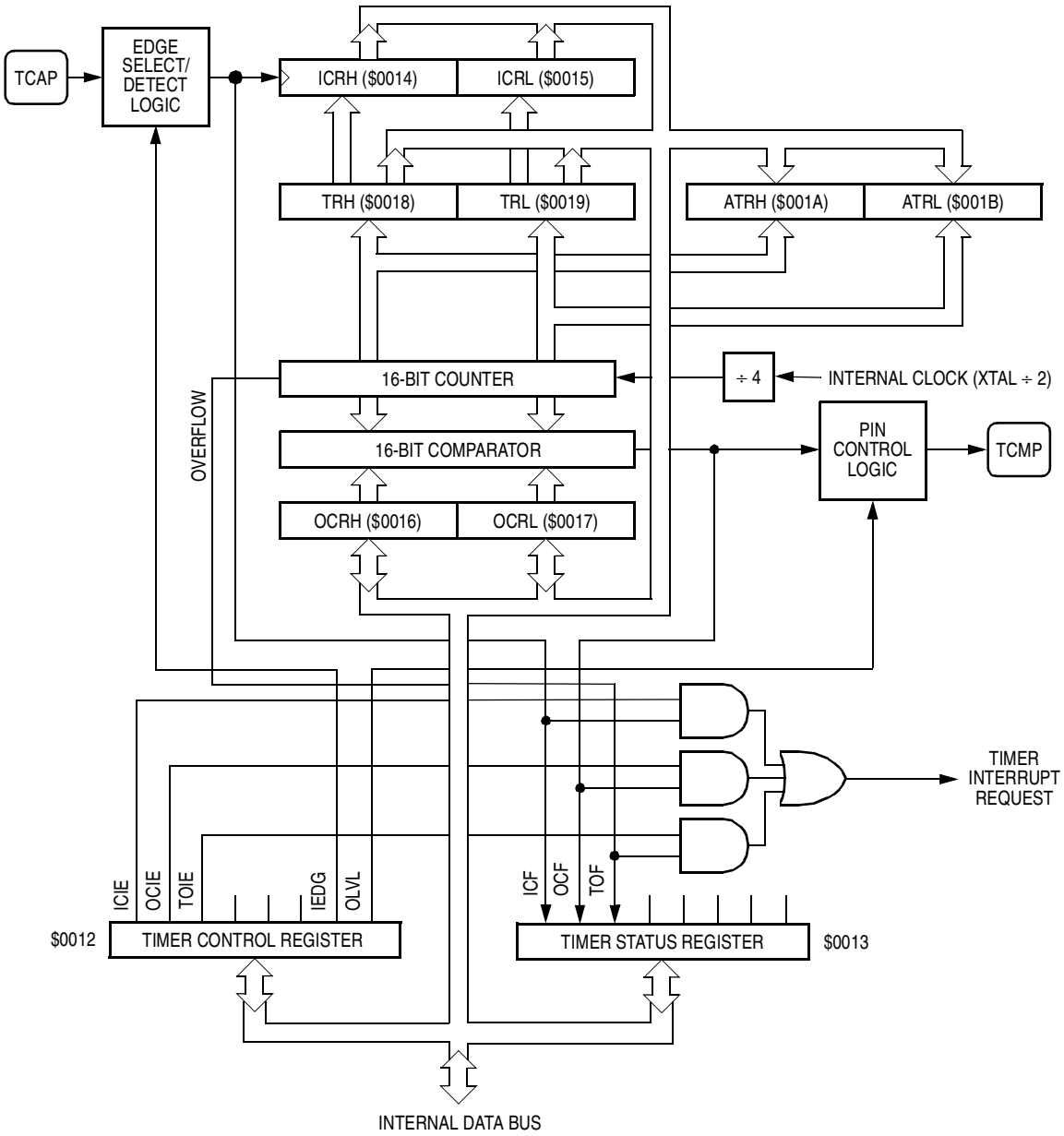


Figure 8-1. Timer Block Diagram

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0012	Timer Control Register (TCR) See page 94.	Read: / Write:	ICIE	OCIE	TOIE	0	0	0	IEDG	OLVL
		Reset:	0	0	0	0	0	0	U	0
$0013	Timer Status Register (TSR) See page 96.	Read:	ICF	OCF	TOF	0	0	0	0	0
		Write:								
		Reset:	U	U	U	0	0	0	0	0
$0014	Input Capture Register High (ICRH) See page 100.	Read:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
		Write:								
		Reset:	Unaffected by reset							
$0015	Input Capture Register Low (ICRL) See page 100.	Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Write:								
		Reset:	Unaffected by reset							
$0016	Output Compare Register High (OCRH) See page 101.	Read: / Write:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
		Reset:	Unaffected by reset							
$0017	Output Compare Register Low (OCRL) See page 101.	Read: / Write:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Reset:	Unaffected by reset							
$0018	Timer Register High (TRH) See page 97.	Read:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
		Write:								
		Reset:	Reset initializes TRH to $FF							
$0019	Timer Register Low (TRL) See page 97.	Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Write:								
		Reset:	Reset initializes TRL to $FC							
$001A	Alternate Timer Register High (ATRH) See page 99.	Read:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
		Write:								
		Reset:	Reset initializes ATRH to $FF							
$001B	Alternate Timer Register Low (ATRL) See page 99.	Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Write:								
		Reset:	Reset initializes ATRL to $FC							

= Unimplemented U = Unaffected

Figure 8-2. Timer I/O Register Summary

Because of the 16-bit timer architecture, the I/O registers for the input capture and output compare functions are pairs of 8-bit registers.

Because the counter is 16 bits long and preceded by a fixed divide-by-four prescaler, the counter rolls over every 262,144 internal clock cycles. Timer resolution with a 4-MHz crystal is 2 µs.

8.3.1 Input Capture

The input capture function can record the time at which an external event occurs. When the input capture circuitry detects an active edge on the input capture pin (TCAP), it latches the contents of the timer registers into the input capture registers. The polarity of the active edge is programmable.

Latching values into the input capture registers at successive edges of the same polarity measures the period of the input signal on the TCAP pin. Latching the counter values at successive edges of opposite polarity measures the pulse width of the signal. **Figure 8-3** shows the logic of the input capture function.

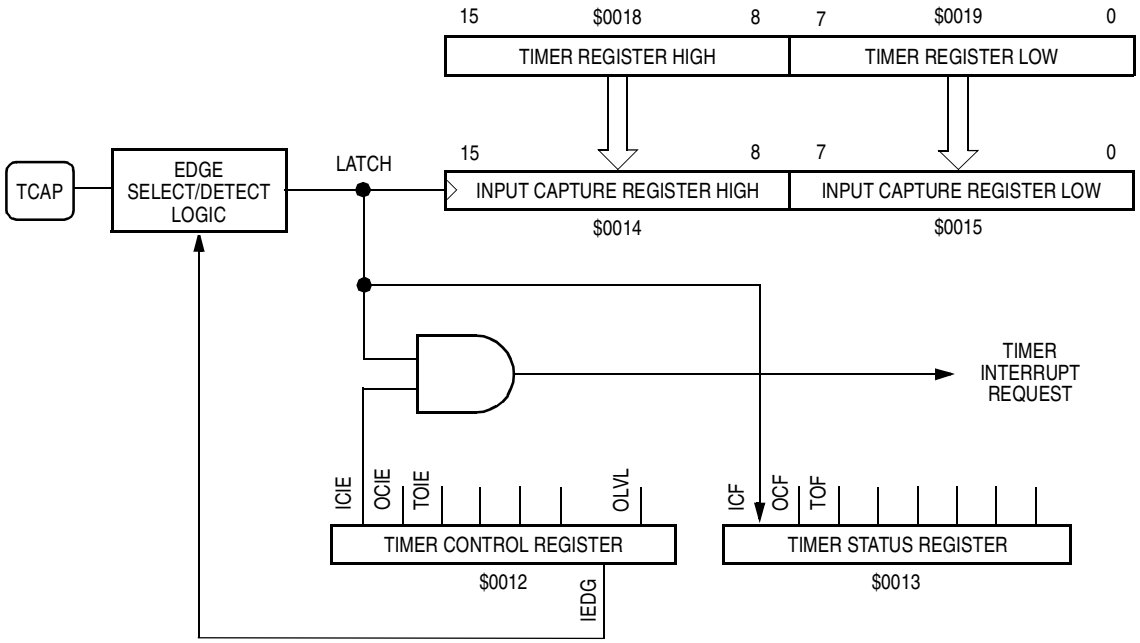


Figure 8-3. Input Capture Operation

8.3.2 Output Compare

The output compare function can generate an output signal when the 16-bit counter reaches a selected value. Software writes the selected value into the output compare registers. On every fourth internal clock cycle the output compare circuitry compares the value of the counter to the value written in the output compare registers. When a match occurs, the timer transfers the programmable output level bit (OLVL) from the timer control register to the output compare pin (TCMP).

Software can use the output compare register to measure time periods, to generate timing delays, or to generate a pulse of specific duration or a pulse train of specific frequency and duty cycle on the TCMP pin. **Figure 8-4** shows the logic of the output compare function.

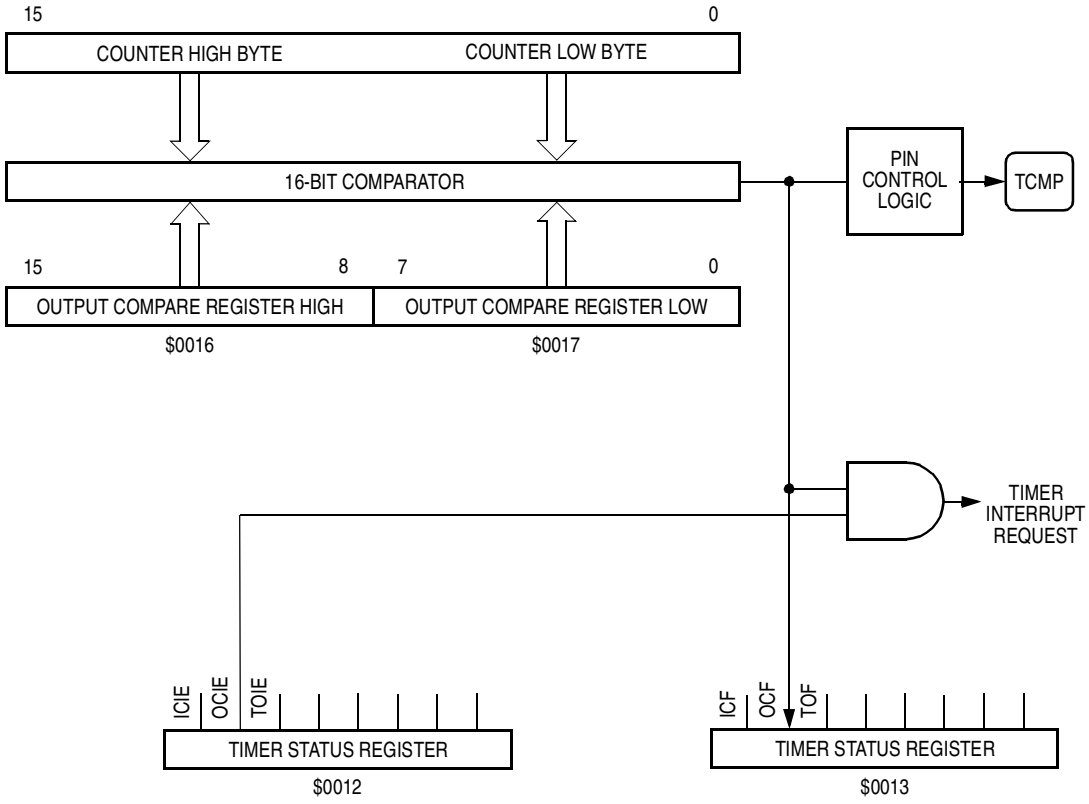


Figure 8-4. Output Compare Operation

8.4 Timer I/O Registers

These registers control and monitor the timer operation:

- Timer control register (TCR)
- Timer status register (TSR)
- Timer registers (TRH and TRL)
- Alternate timer registers (ATRH and ATRL)
- Input capture registers (ICRH and ICRL)
- Output compare registers (OCRH and OCRL)

8.4.1 Timer Control Register

The timer control register (TCR) as shown in **Figure 8-5** performs these functions:

- Enables input capture interrupts
- Enables output compare interrupts
- Enables timer overflow interrupts
- Controls the active edge polarity of the TCAP signal
- Controls the active level of the TCMP output

Address: $0012

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	ICIE	OCIE	TOIE	0	0	0	IEDG	OLVL
Reset:	0	0	0	0	0	0	U	0

U = Unaffected

Figure 8-5. Timer Control Register (TCR)

ICIE — Input Capture Interrupt Enable Bit

This read/write bit enables interrupts caused by an active signal on the TCAP pin. Reset clears the ICIE bit.

1 = Input capture interrupts enabled
0 = Input capture interrupts disabled

OCIE — Output Compare Interrupt Enable Bit

This read/write bit enables interrupts caused by an active signal on the TCMP pin. Reset clears the OCIE bit.

1 = Output compare interrupts enabled
0 = Output compare interrupts disabled

TOIE — Timer Overflow Interrupt Enable Bit

This read/write bit enables interrupts caused by a timer overflow. Reset clears the TOIE bit.

1 = Timer overflow interrupts enabled
0 = Timer overflow interrupts disabled

IEDG — Input Edge Bit

The state of this read/write bit determines whether a positive or negative transition on the TCAP pin triggers a transfer of the contents of the timer register to the input capture registers. Reset has no effect on the IEDG bit.

1 = Positive edge (low-to-high transition) triggers input capture
0 = Negative edge (high-to-low transition) triggers input capture

OLVL — Output Level Bit

The state of this read/write bit determines whether a logic 1 or a logic 0 appears on the TCMP pin when a successful output compare occurs. Reset clears the OLVL bit.

1 = TCMP goes high on output compare
0 = TCMP goes low on output compare

Bits 4–2 — Not used; these bits always read 0

8.4.2 Timer Status Register

The timer status register (TSR) is a read-only register shown in **Figure 8-6** contains flags for these events:

- An active signal on the TCAP pin, transferring the contents of the timer registers to the input capture registers
- A match between the 16-bit counter and the output compare registers, transferring the OLVL bit to the TCMP pin
- A timer rollover from $FFFF to $0000

Address: $0013

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	ICF	OCF	TOF	0	0	0	0	0
Write:								
Reset:	U	U	U	0	0	0	0	0

= Unimplemented U = Unaffected

Figure 8-6. Timer Status Register (TSR)

ICF — Input Capture Flag

The ICF bit is set automatically when an edge of the selected polarity occurs on the TCAP pin. Clear the ICF bit by reading the timer status register with ICF set and then reading the low byte ($0015) of the input capture registers. Reset has no effect on ICF.

1 = Input capture
0 = No input capture

OCF — Output Compare Flag

The OCF bit is set automatically when the value of the timer registers matches the contents of the output compare registers. Clear the OCF bit by reading the timer status register with OCF set and then reading the low byte ($0017) of the output compare registers. Reset has no effect on OCF.

1 = Output compare
0 = No output compare

TOF — Timer Overflow Flag

The TOF bit is automatically set when the 16-bit counter rolls over from $FFFF to $0000. Clear the TOF bit by reading the timer status register with TOF set and then reading the low byte ($0019) of the timer registers. Reset has no effect on TOF.

1 = Timer overflow
0 = No timer overflow

Bits 4–0 — Not used; these bits always read 0

8.4.3 Timer Registers

The read-only timer registers (TRH and TRL) shown in **Figure 8-7** contain the current high and low bytes of the 16-bit counter. Reading TRH before reading TRL causes TRL to be latched until TRL is read. Reading TRL after reading the timer status register clears the timer overflow flag bit (TOF). Writing to the timer registers has no effect.

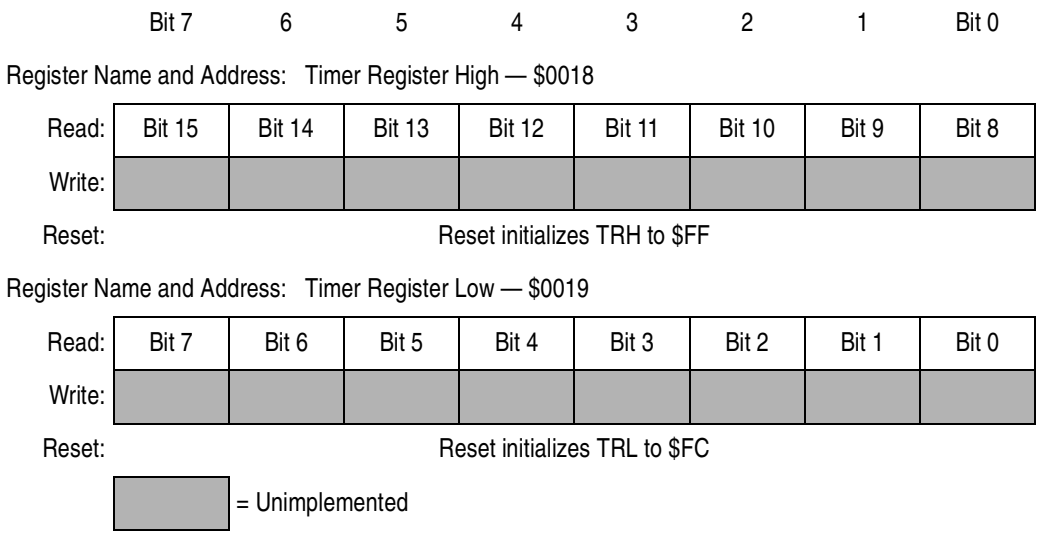


Figure 8-7. Timer Registers (TRH and TRL)

Reading TRH returns the current value of the high byte of the counter and causes the low byte to be latched into a buffer, as shown in **Figure 8-8**. The buffer value remains fixed even if the high byte is read more than once. Reading TRL reads the transparent low byte buffer and completes the read sequence of the timer registers.

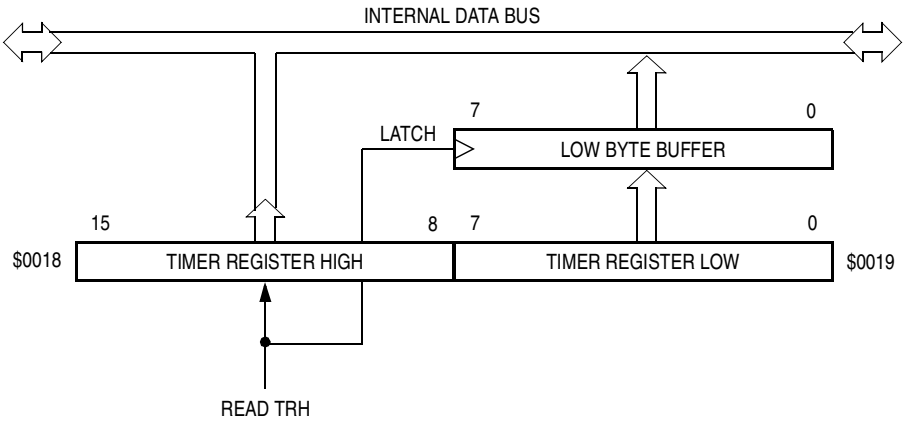


Figure 8-8. Timer Register Reads

NOTE: *To prevent interrupts from occurring between readings of TRH and TRL, set the interrupt mask (I bit) in the condition code register before reading TRH, and clear the mask after reading TRL.*

8.4.4 Alternate Timer Registers

The alternate timer registers (ATRH and ATRL) shown in **Figure 8-9** contain the current high and low bytes of the 16-bit counter. Reading ATRH before reading ATRL causes ATRL to be latched until ATRL is read. Reading does not affect the timer overflow flag (TOF). Writing to the alternate timer registers has no effect.

	Bit 7	6	5	4	3	2	1	Bit 0
Register Name and Address: Alternate Timer Register High — $001A								
Read:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Write:								
Reset:	Reset initializes ATRH to $FF							
Register Name and Address: Alternate Timer Register Low — $001B								
Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Write:								
Reset:	Reset initializes ATRL to $FC							

= Unimplemented

Figure 8-9. Alternate Timer Registers (ATRH and ATRL)

Reading ATRH returns the current value of the high byte of the counter and causes the low byte to be latched into a buffer, as shown in **Figure 8-10**.

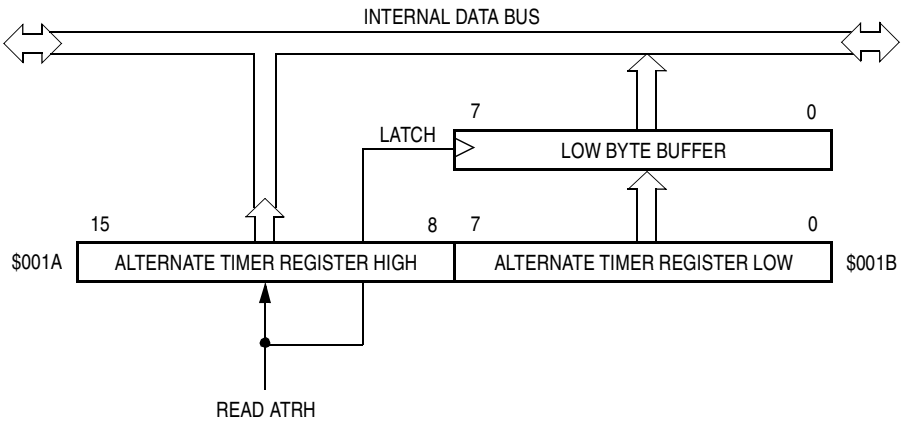


Figure 8-10. Alternate Timer Register Reads

NOTE: *To prevent interrupts from occurring between readings of ATRH and ATRL, set the interrupt mask (I bit) in the condition code register before reading ATRH, and clear the mask after reading ATRL.*

8.4.5 Input Capture Registers

When a selected edge occurs on the TCAP pin, the current high and low bytes of the 16-bit counter are latched into the read-only input capture registers (ICRH and ICRL) shown in **Figure 8-11**. Reading ICRH before reading ICRL inhibits further captures until ICRL is read. Reading ICRL after reading the timer status register clears the input capture flag (ICF). Writing to the input capture registers has no effect.

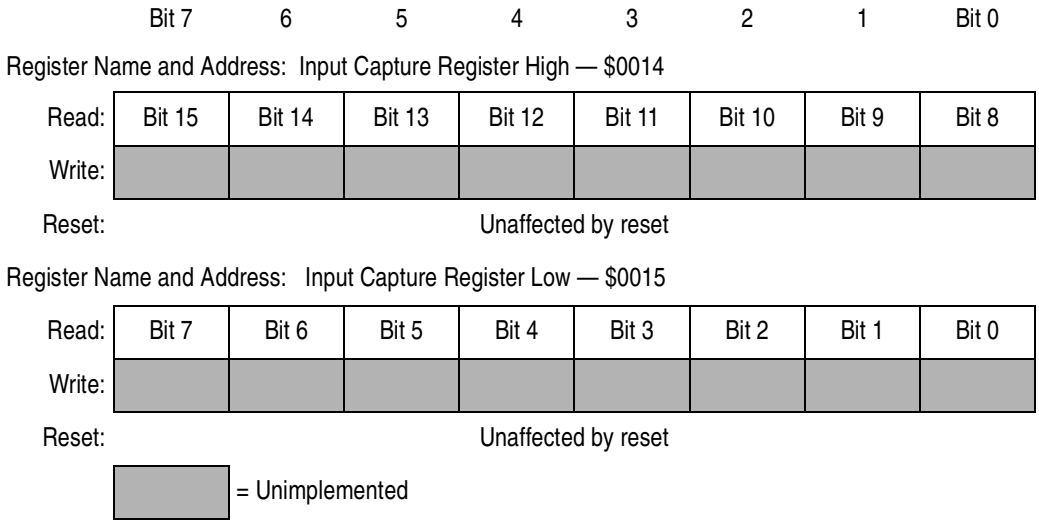


Figure 8-11. Input Capture Registers (ICRH and ICRL)

NOTE: *To prevent interrupts from occurring between readings of ICRH and ICRL, set the interrupt mask (I bit) in the condition code register before reading ICRH and clear the mask after reading ICRL.*

8.4.6 Output Compare Registers

When the value of the 16-bit counter matches the value in the read/write output compare registers (OCRH and OCRL) shown in **Figure 8-12**, the planned TCMP pin action takes place. Writing to OCRH before writing to OCRL inhibits timer compares until OCRL is written. Reading or writing to OCRL after reading the timer status register clears the output compare flag (OCF).

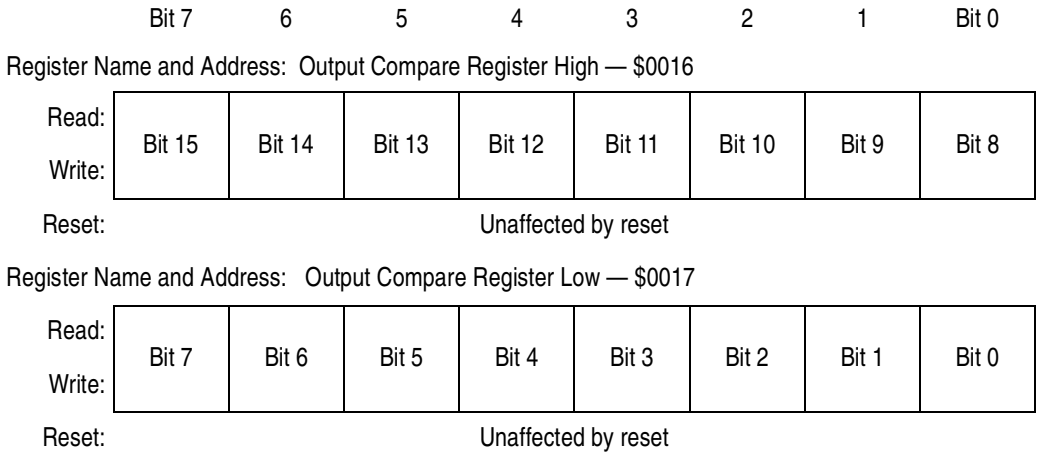


Figure 8-12. Output Compare Registers (OCRH and OCRL)

To prevent OCF from being set between the time it is read and the time the output compare registers are updated, use this procedure:

1. Disable interrupts by setting the I bit in the condition code register.
2. Write to OCRH. Compares are now inhibited until OCRL is written.
3. Clear bit OCF by reading the timer status register (TSR).
4. Enable the output compare function by writing to OCRL.
5. Enable interrupts by clearing the I bit in the condition code register.

Section 9. EPROM/OTPROM (PROM)

9.1 Contents

9.2 Introduction

This section describes erasable, programmable read-only memory/one-time programmable read-only memory (EPROM/OTPROM (PROM)) programming.

9.3 EPROM/OTPROM (PROM) Programming

The internal PROM can be programmed efficiently using the Motorola MC68HC05PGMR-2 programmer board, which can be purchased from a Motorola-authorized distributor. The user can program the microcontroller unit (MCU) using this printed circuit board (PCB) in conjunction with an EPROM device already programmed with user code.

Only standalone programming is discussed in this section. For more information concerning the MC68HC05PGMR and its usages, contact a local Motorola representative for a copy of the *MC68HC05PGMR Programmer Board User's Manual #2*, Motorola document number MC68HC05PGMR2/D1.

Refer to **Figure 9-1** for an EPROM programming flowchart. **Figure 9-2** provides a schematic of the MC68HC05PGMR PCB with the reference designators defined in **Table 9-1**.

Table 9-1. MC68HC05PGMR PCB Reference Designators

Reference Designators	Device Type	Ground	+5 V	+12 V	–12 V	V_{PP}	Notes
U1	2764	14, 20	1, 26, 27, 28	—	—	—	8 K x 8-bit EPROM
U2	MCU	20	40	—	—	3	40-pin DIP socket
U3	MCU	22	44	—	—	4	44-lead PLCC socket
U4	MC145406	9	16	1	8	—	Driver/receiver
VR1	NMA0512S	2.5	1	6	4	—	DC-DC converter

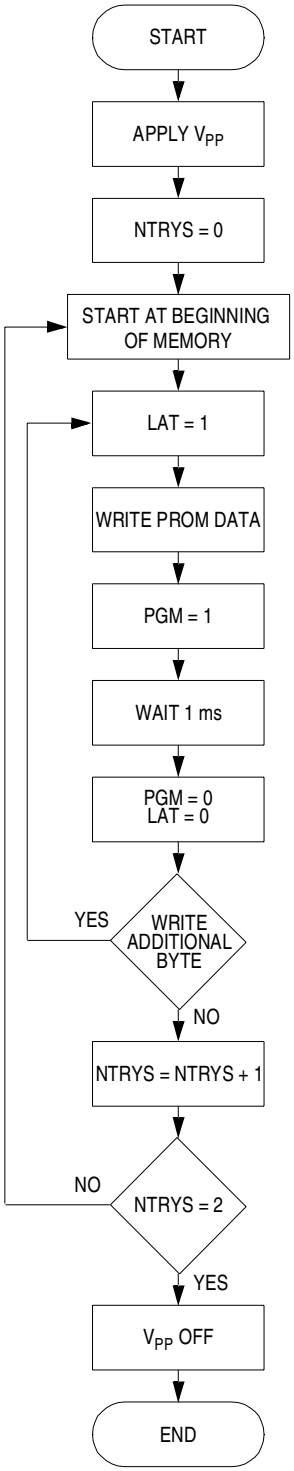
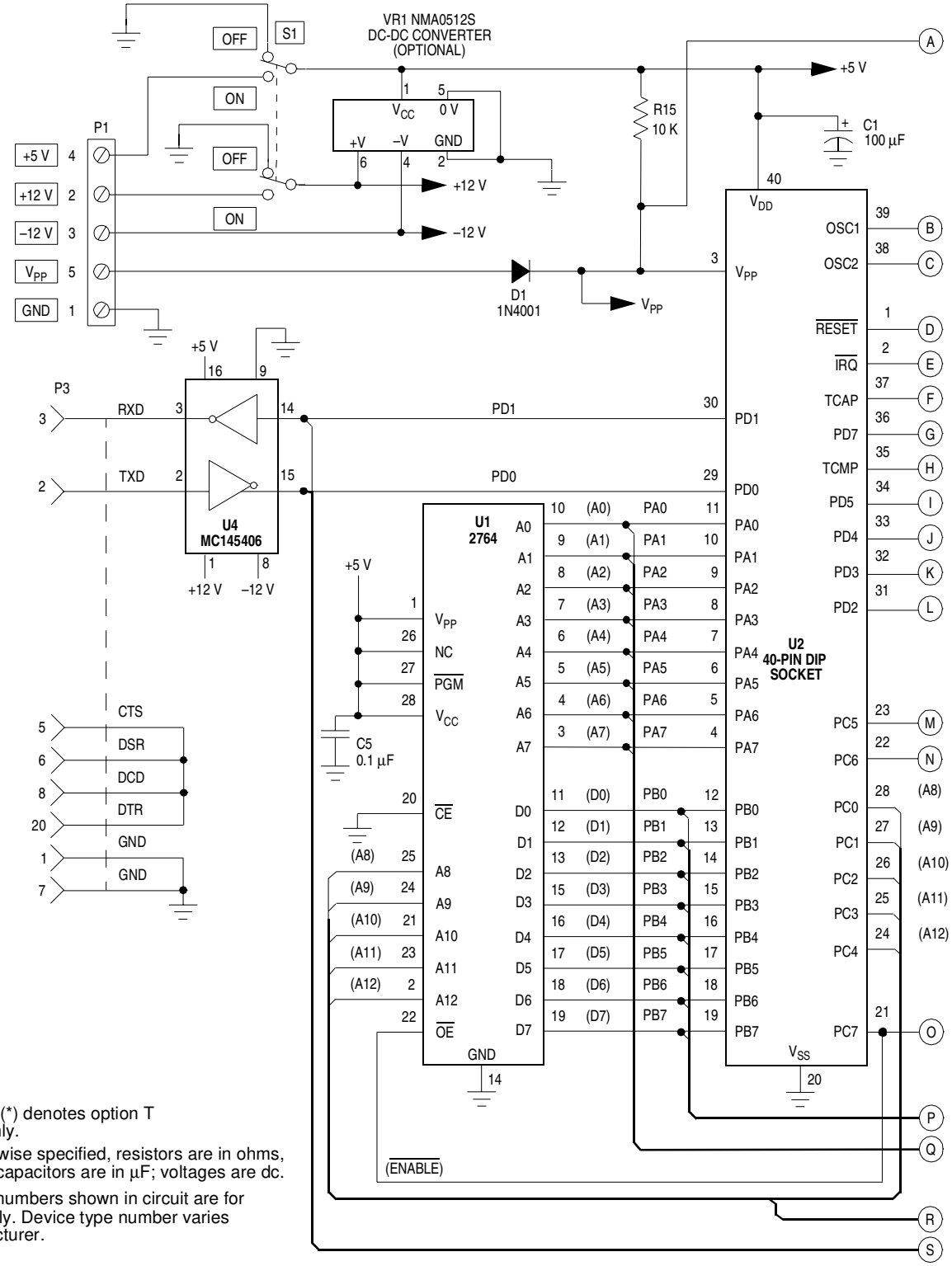


Figure 9-1. EPROM/OTPROM Programming Flowchart



Notes:

1. The asterisk (*) denotes option T command only.
2. Unless otherwise specified, resistors are in ohms, ±5% 1/4 W; capacitors are in µF; voltages are dc.
3. Device type numbers shown in circuit are for reference only. Device type number varies with manufacturer.

Figure 9-2. PROM Programming Circuit

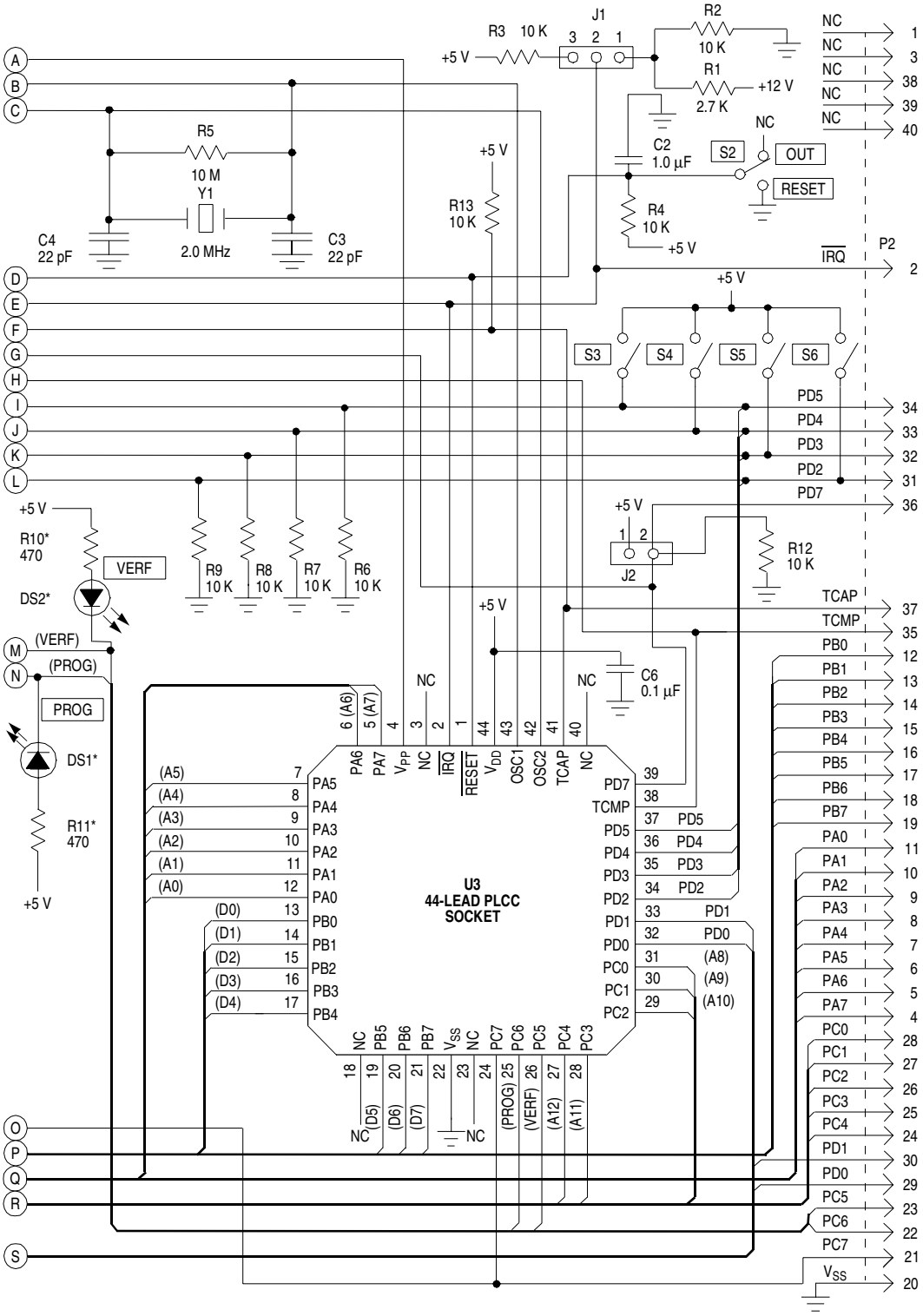


Figure 9-2. PROM Programming Circuit (Continued)

To program the PROM MCU, the MCU is installed in the PCB, along with an EPROM device programmed with user code; the MCU is then subjected to a series of routines. The routines necessary to program, verify, and secure the PROM MCU are:

- Program and verify PROM
- Verify PROM contents only
- Secure PROM and verify
- Secure PROM and dump through the serial communications interface (SCI)

Other board routines available to the user are:

- Load program into random-access memory (RAM) and execute
- Execute program in RAM
- Dump PROM contents (binary upload)

The user first configures the MCU for the bootstrap mode of operations by installing a fabricated jumper across pins 1 and 2 of the board's mode select header, J1. Next, the board's mode switches (S3, S4, S5, and S6) are set to determine the routine to be executed after the next reset, as shown in **Table 9-2**.

Table 9-2. PROM Programming Routines

Routine	S3	S4	S5	S6
Program and verify PROM	Off	Off	Off	Off
Verify PROM contents only	Off	Off	On	Off
Secure PROM contents and verify	On	Off	On	Off
Secure PROM contents and dump	On	On	On	Off
Load program into RAM and execute	Off	On	Off	Off
Execute program in RAM	Off	Off	Off	On
Dump PROM contents	Off	On	On	Off

9.3.1 Program Register

The program register (PROG) shown in **Figure 9-3** is used for PROM programming.

Address: $001C

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	0	0	0	0	0	LAT	0	PGM
Reset:	0	0	0	0	0	0	0	0

Figure 9-3. Program Register (PROG)

LAT — Latch Enable Bit

This bit is both readable and writable.

1 = Enables PROM data and address bus latches for programming on the next byte write cycle

0 = Latch disabled. PROM data and address buses are unlatched for normal CPU operations.

PGM — Program Bit

If LAT is cleared, PGM cannot be set.

1 = Enables V_{PP} power to the PROM for programming

0 = V_{PP} is disabled.

Bits 1 and 3–7 — Not used; always read 0

9.3.2 Preprogramming Steps

Before programming the PROM using an MC68HC05PGMR PCB in standalone mode, the user should ensure that:

- A jumper is installed on pins 1 and 2 of mode select header J1.
- An EPROM is programmed with the necessary user code.
- The erasure window (if any) of the device to be programmed is covered.
- V_{DD} of +5 Vdc is available on the board.
- V_{PP} is available on the board.

NOTE: *If the V_{PP} level at the MCU exceeds +16 Vdc, then the MC68HC705C8A MCU device will suffer permanent damage.*

Once those conditions are met, the user should take these steps before beginning programming:

1. Remove the V_{PP} power source.
2. Set switch 1 in the OFF position (removes V_{DD}).
3. Place the programmed EPROM in socket U1.
4. Insert the erased PROM MCU device to be programmed in the proper socket:
   - MC68HC705C8S or MC68HC705C8P in socket U2 (40-pin dual in-line package (DIP)) or
   - MC68HC705C8FN in socket U3 (44-pin plastic leaded chip carrier (PLCC)) with the device notch at the upper right corner of the socket.
5. Set switch S2 in the RESET position.

NOTE: *No PROM MCU should be inserted in or removed from its board socket (U2 or U3) while V_{PP} (P1, slot 5) or V_{DD} (switch 1) is active on the board.*

9.4 PROM Programming Routines

This subsection describes the routines necessary to program, verify, and secure the PROM device, and other routines available to the user.

9.4.1 Program and Verify PROM

The program and verify PROM routine copies the contents of the external EPROM into the MCU PROM with direct correspondence between the addresses. Memory addresses in the MCU that are not implemented in PROM are skipped. Unprogrammed addresses in the EPROM being copied should contain $00 bytes to speed up the programming process.

To run the program and verify the PROM routine on the PROM MCU, take these steps:

1. Set switch 1 in the ON position (restores V_{DD}).
2. Restore the V_{PP} power source.
3. Set switches S3, S4, S5, and S6 in the OFF position (selects proper routine).
4. Set switch 2 in the OUT position (routine is activated).

   The red light-emitting diode (LED) is illuminated, showing that the programming part of the routine is running. The LED goes out when programming is finished. The verification part of the routine now begins. When the green LED is illuminated, verification is successfully completed and the routine is finished.
5. Set switch 2 in the RESET position.

At this point, if no other MCU is to be programmed or secured, remove V_{PP} power from the board. If another routine is to be performed on the MCU being programmed, the user can then set switches S3, S4, S5, and S6 to the positions necessary to select the next routine, and begin the routine by setting switch 2 to the OUT position. If no other routine is to be performed, remove V_{DD} from the board and remove the MCU from the programming socket.

9.4.2 Verify PROM Contents

The verify PROM contents routine is normally run automatically after the PROM is programmed. Direct entry to this routine causes the PROM contents of the MCU to be compared to the contents of the external memory locations of the EPROM at the same addresses.

To invoke the verify PROM contents routine of the MCU, take these steps:

1. Set switch 1 in the ON position (restores V_{DD}).
2. Connect V_{PP} to V_{DD}.
3. Set switches S3, S4, and S6 in the OFF position.
4. Set S5 in the ON position.
5. Set switch 2 in the OUT position (routine is activated).

   The red LED is not illuminated during this routine, since no programming takes place. If verification fails, the routine halts with the failing address in the external memory bus. When the green LED is illuminated, verification is completed successfully and the routine is finished.
6. Set switch 2 in the RESET position.

At this point, if another routine is to be performed on the MCU being programmed, the user can set switches S3, S4, S5, and S6 to the positions necessary to select the next routine and move switch S2 to the OUT position to start the routine. If no other routine is to be performed, remove V_{DD} from the board and remove the MCU from the programming socket.

9.4.3 Secure PROM

The secure PROM routines are used after the PROM is successfully programmed and verified. Only the SEC bit of the option register ($1FDF) is programmed, but V_{PP} is necessary. Once this bit is programmed, PROM is secure and can be neither verified nor dumped.

9.4.4 Secure PROM and Verify

This routine is used after the PROM is programmed successfully to verify the contents of the MCU PROM against the contents of the EPROM and then to secure the PROM. To accomplish this routine, take these steps:

1. Set switch 1 in the ON position (restores V_{DD}).
2. Restore V_{PP} power to the programming board.
3. Set switches S4 and S6 in the OFF position.
4. Set switches S3 and S5 in the ON position.
5. Set switch 2 in the OUT position (routine is activated). Execution time for this routine is about one second.
6. Set switch 2 in the RESET position when the routine is completed.

No LED is illuminated during this routine. Further, the end of the routine does not mean that the SEC bit was verified. To ensure that security is properly enabled, attempt to perform another verify routine. If the green LED does not light, the PROM has been secured properly.

9.4.5 Secure PROM and Dump

This routine is used after the PROM is successfully programmed to dump the contents of the MCU PROM through the SCI (binary upload) and then to secure the PROM. To accomplish this routine, take these steps:

1. Set switch 1 in the ON position (restores V_{DD}).
2. Restore V_{PP} power to the programming board.
3. Set switch S6 in the OFF position.
4. Set switches S3, S4, and S5 in the ON position.
5. Set switch 2 in the OUT position (routine is activated). Execution time for this routine is about one second.
6. Set switch 2 in the RESET position when the routine is completed.

No LED is illuminated during this routine. Further, the end of the routine does not mean that the SEC bit was verified. To ensure that security is properly enabled, attempt to perform another verify routine. If the green LED does not light, the PROM has been secured properly.

9.4.6 Load Program into RAM and Execute

In the load program in RAM and execute routine, user programs are loaded via the SCI port and then executed. Data is loaded sequentially starting at address $0050. After the last byte is loaded, control is transferred to the RAM program starting at $0051. The first byte loaded is the count of the total number of bytes in the program plus the count byte. The program starts at location $0051 in RAM. During initialization, the SCI is configured for eight data bits and one stop bit. The baud rate is 4800 with a 2-MHz crystal or 9600 with a 4-MHz crystal.

To load a program into RAM and execute it, take these steps:

1. Set switch 1 in the ON position (restores V_{DD}).
2. Connect V_{PP} to V_{DD}.
3. Set switches S3, S5, and S6 in the OFF position.
4. Set switch S4 in the ON position.
5. Set switch 2 in the OUT position (routine is activated).

The downloaded program starts executing as soon as the last byte is received by the SCI.

Execution of the routine can be held off by setting the byte count in the count byte (the first byte loaded) to a value greater than the number of bytes to be loaded. After loading the last byte, the firmware waits for more data. Program execution does not begin. At this point, placing switch 2 in the RESET position resets the MCU with the RAM data intact. Any other routine can be entered, including the one to execute the program in RAM, simply by setting switches S3–S6 as necessary to select the desired routine, then setting switch 2 in the OUT position.

9.4.7 Execute Program in RAM

This routine allows the MCU to transfer control to a program previously loaded in RAM. This program is executed once bootstrap mode is entered, if switch S6 is in the ON position and switch 2 is in the OUT position, without any firmware initialization. The program must start at location $0051 to be compatible with the load program in RAM routine.

To run the execute program in RAM routine, take these steps:

1. Set switch 1 in the ON position (restores V_{DD}).
2. Connect V_{PP} to V_{DD}.
3. Set switch S6 in the OFF position.
4. Switches S3, S4, and S5 can be in either position.
5. Set switch 2 in the OUT position (routine is activated).

NOTE: *The non-programmable watchdog COP is disabled in bootloader mode, even if the NCOPE bit is programmed.*

9.4.8 Dump PROM Contents

In the dump PROM contents routine, the PROM contents are dumped sequentially to the SCI output, provided the PROM has not been secured. The first location sent is $0020 and the last location sent is $1FFF. Unused locations are skipped so that no gaps exist in the data stream. The external memory address lines indicate the current location being sent. Data is sent with eight data bits and one stop bit at 4800 baud with a 2-MHz crystal or 9600 baud with a 4-MHz crystal.

To run the dump PROM contents routine, take these steps:

1. Set switch 1 in the ON position (restores V_{DD}).
2. Connect V_{PP} to V_{DD}.
3. Set switches S3 and S6 in the OFF position.
4. Set switches S4 and S5 in the ON position.
5. Set switch 2 in the OUT position (routine is activated).
6. Once PROM dumping is complete, set switch 2 in the RESET position.

9.5 Control Registers

This subsection describes the three registers that control memory configuration, PROM security, and IRQ edge or level sensitivity; port B pullups; and non-programmable COP enable/disable.

9.5.1 Option Register

The option register shown in **Figure 9-4** is used to select the IRQ sensitivity, enable the PROM security, and select the memory configuration.

Address: $1FDF

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	RAM0	RAM1	0	0	SEC*		IRQ	0
Reset:	0	0	0	0	*	U	1	0

*Implemented as an EPROM cell

= Unimplemented U = Unaffected

Figure 9-4. Option Register (Option)

RAM0 — Random-Access Memory Control Bit 0

1 = Maps 32 bytes of RAM into page zero starting at address $0030. Addresses from $0020 to $002F are reserved. This bit can be read or written at any time, allowing memory configuration to be changed during program execution.

0 = Provides 48 bytes of PROM at location $0020–$005F.

RAM1 — Random-Access Memory Control Bit 1

1 = Maps 96 bytes of RAM into page one starting at address $0100. This bit can be read or written at any time, allowing memory configuration to be changed during program execution.

0 = Provides 96 bytes of PROM at location $0100.

SEC — Security Bit

This bit is implemented as an EPROM cell and is not affected by reset.

1 = Security enabled
0 = Security off; bootloader able to be enabled

IRQ — Interrupt Request Pin Sensitivity Bit

IRQ is set only by reset, but can be cleared by software. This bit can only be written once.

1 = $\overline{\text{IRQ}}$ pin is both negative edge- and level-sensitive.
0 = $\overline{\text{IRQ}}$ pin is negative edge-sensitive only.

Bits 5, 4, and 0 — Not used; always read 0

Bit 2 — Unaffected by reset; reads either 1 or 0

9.5.2 Mask Option Register 1

Mask option register 1 (MOR1) shown in **Figure 9-5** is an EPROM register that enables the port B pullup devices. Data from MOR1 is latched on the rising edge of the voltage on the $\overline{\text{RESET}}$ pin.
See **4.3.3 Port B Interrupts**.

Address: $1FF0

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	PBPU7	PBPU6	PBPU5	PBPU4	PBPU3	PBPU2	PBPU1	PBPU0/ COPC
Reset:	Unaffected by reset							
Erased:	0	0	0	0	0	0	0	0

Figure 9-5. Mask Option Register 1 (MOR1)

PBPU7–PBPU0/COPC — Port B Pullup Enable Bits 7–0

These EPROM bits enable the port B pullup devices.

1 = Port B pullups enabled
0 = Port B pullups disabled

NOTE: *PBPU0/COPC programmed to a 1 enables the port B pullup bit. This bit is also used to clear the non-programmable COP (MC68HC05C4A type). Writing to this bit to clear the COP will not affect the state of the port B pull-up (bit 0). See* ***5.3.3 Programmable and Non-Programmable COP Watchdog Resets****.*

When using the MC68HC705C8A in an MC68HC705C8 or MC68HSC705C8 application, program locations $1FF0 and $1FF1 to $00.

9.5.3 Mask Option Register 2

Mask option register 2 (MOR2) shown in **Figure 9-6** is an EPROM register that enables the non-programmable COP watchdog. Data from MOR2 is latched on the rising edge of the voltage on the $\overline{\text{RESET}}$ pin. See **5.3.3 Programmable and Non-Programmable COP Watchdog Resets**.

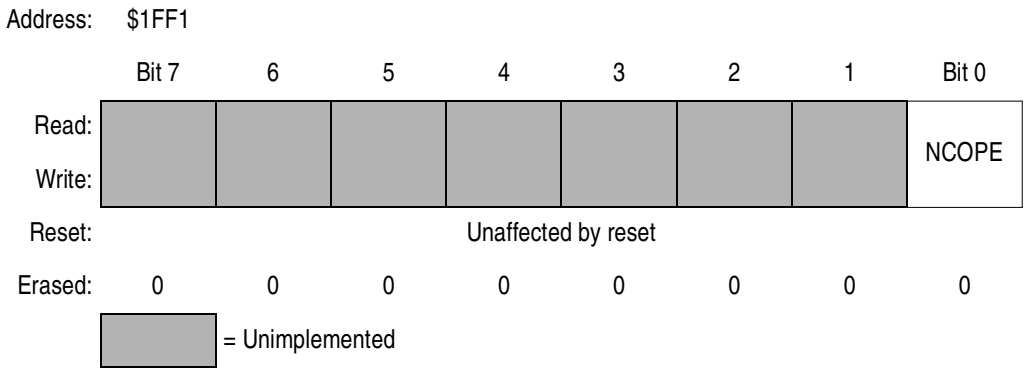


Figure 9-6. Mask Option Register 2 (MOR2)

NCOPE — Non-Programmable COP Watchdog Enable Bit

This EPROM bit enables the non-programmable COP watchdog.

1 = Non-programmable COP watchdog enabled

0 = Non-programmable COP watchdog disabled

9.6 EPROM Erasing

The erased state of an EPROM or OTPROM byte is $00. EPROM devices can be erased by exposure to a high intensity ultraviolet (UV) light with a wave length of 2537 Å. The recommended erasure dosage (UV intensity on a given surface area x exposure time) is 15 Ws/cm^2. UV lamps should be used without short-wave filters, and the EPROM device should be positioned about one inch from the UV source.

OTPROM devices are shipped in an erased state. Once programmed, they cannot be erased. Electrical erasing procedures cannot be performed on either EPROM or OTPROM devices.

EPROM/OTPROM (PROM)

Section 10. Serial Communications Interface (SCI)

10.1 Contents

10.2 Introduction

The serial communications interface (SCI) module allows high-speed asynchronous communication with peripheral devices and other microcontroller units (MCUs).

10.3 Features

Features of the SCI module include:

- Standard mark/space non-return-to-zero format
- Full-duplex operation
- 32 programmable baud rates
- Programmable 8-bit or 9-bit character length
- Separately enabled transmitter and receiver
- Two receiver wakeup methods:
  - Idle line wakeup
  - Address mark wakeup
- Interrupt-driven operation capability with five interrupt flags:
  - Transmitter data register empty
  - Transmission complete
  - Receiver data register full
  - Receiver overrun
  - Idle receiver input
- Receiver framing error detection
- 1/16 bit-time noise detection

10.4 SCI Data Format

The SCI uses the standard non-return-to-zero mark/space data format illustrated in **Figure 10-1**.

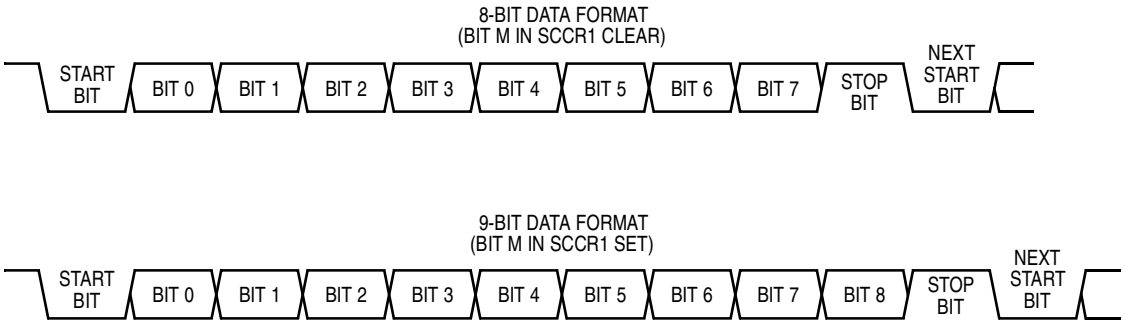


Figure 10-1. SCI Data Format

10.5 SCI Operation

The SCI allows full-duplex, asynchronous, RS232 or RS422 serial communication between the MCU and remote devices, including other MCUs. The transmitter and receiver of the SCI operate independently, although they use the same baud-rate generator. This subsection describes the operation of the SCI transmitter and receiver.

10.5.1 Transmitter

Figure 10-2 shows the structure of the SCI transmitter. **Figure 10-3** is a summary of the SCI transmitter input/output (I/O) registers.

- Character Length — The transmitter can accommodate either 8-bit or 9-bit data. The state of the M bit in SCI control register 1 (SCCR1) determines character length. When transmitting 9-bit data, bit T8 in SCCR1 is the ninth bit (bit 8).
- Character Transmission — During transmission, the transmit shift register shifts a character out to the PD1/TDO pin. The SCI data register (SCDR) is the write-only buffer between the internal data bus and the transmit shift register.

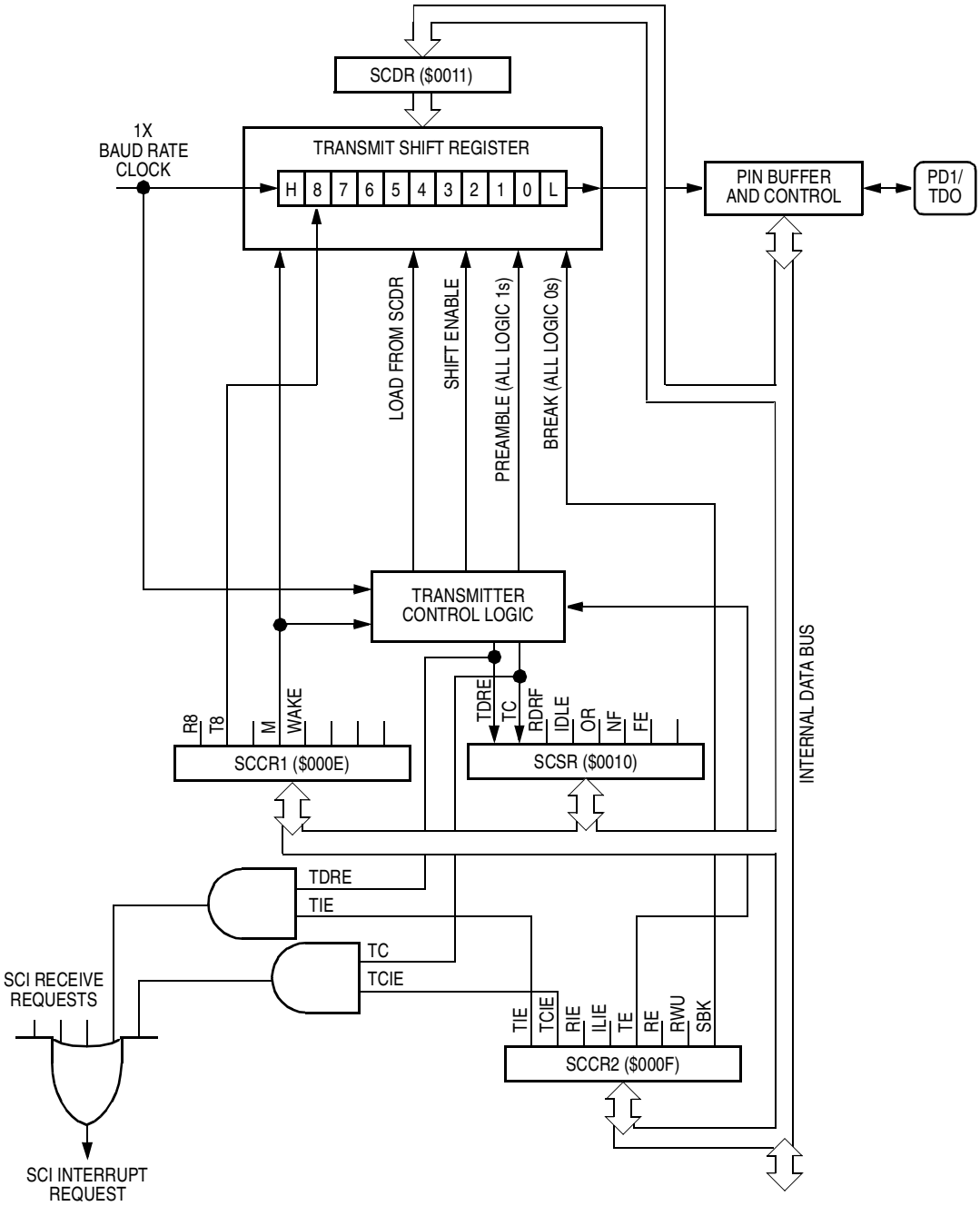


Figure 10-2. SCI Transmitter

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$000D	Baud Rate Register (Baud) See page 136.	Read:			SCP1	SCP0		SCR2	SCR1	SCR0
		Write:								
		Reset:	U	U	0	0	U	U	U	U
$000E	SCI Control Register 1 (SCCR1) See page 130.	Read:	R8	T8		M	WAKE			
		Write:								
		Reset:	U	U		U	U			
$000F	SCI Control Register 2 (SCCR2) See page 131.	Read:	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0010	SCI Status Register (SCSR) See page 133.	Read:	TDRE	TC	RDRF	IDLE	OR	NF	FE	
		Write:								
		Reset:	1	1	0	0	0	0	0	U
$0011	SCI Data Register (SCDR) See page 129.	Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Write:								
		Reset:	Unaffected by reset							

= Unimplemented U = Unaffected

Figure 10-3. SCI Transmitter I/O Register Summary

Writing a logic 1 to the TE bit in SCI control register 2 (SCCR2) and then writing data to the SCDR begins the transmission. At the start of a transmission, transmitter control logic automatically loads the transmit shift register with a preamble of logic 1s. After the preamble shifts out, the control logic transfers the SCDR data into the shift register. A logic 0 start bit automatically goes into the least significant bit (LSB) position of the shift register, and a logic 1 stop bit goes into the most significant bit (MSB) position.

When the data in the SCDR transfers to the transmit shift register, the transmit data register empty (TDRE) flag in the SCI status register (SCSR) becomes set. The TDRE flag indicates that the SCDR can accept new data from the internal data bus.

When the shift register is not transmitting a character, the PD1/TDO pin goes to the idle condition, logic 1. If software clears the TE bit during the idle condition, and while TDRE is set, the transmitter relinquishes control of the PD1/TDO pin.

- Break Characters — Writing a logic 1 to the SBK bit in SCCR2 loads the shift register with a break character. A break character contains all logic 0s and has no start and stop bits. Break character length depends on the M bit in SCCR1. As long as SBK is at logic 1, transmitter logic continuously loads break characters into the shift register. After software clears the SBK bit, the shift register finishes transmitting the last break character and then transmits at least one logic 1. The automatic logic 1 at the end of a break character is to guarantee the recognition of the start bit of the next character.
- Idle Characters — An idle character contains all logic 1s and has no start or stop bits. Idle character length depends on the M bit in SCCR1. The preamble is a synchronizing idle character that begins every transmission.

  Clearing the TE bit during a transmission relinquishes the PD1/TDO pin after the last character to be transmitted is shifted out. The last character may already be in the shift register, or waiting in the SCDR, or it may be a break character generated by writing to the SBK bit. Toggling TE from logic 0 to logic 1 while the last character is in transmission generates an idle character (a preamble) that allows the receiver to maintain control of the PD1/TDO pin.
- Transmitter Interrupts — These sources can generate SCI transmitter interrupt requests:
  - Transmit Data Register Empty (TDRE) — The TDRE bit in the SCSR indicates that the SCDR has transferred a character to the transmit shift register. TDRE is a source of SCI interrupt requests. The transmission complete interrupt enable bit (TCIE) in SCCR2 is the local mask for TDRE interrupts.
  - Transmission Complete (TC) — The TC bit in the SCSR indicates that both the transmit shift register and the SCDR are empty and that no break or idle character has been generated. TC is a source of SCI interrupt requests. The transmission complete interrupt enable bit (TCIE) in SCCR2 is the local mask for TC interrupts.

10.5.2 Receiver

Figure 10-4 shows the structure of the SCI receiver. Refer to **Figure 10-3** for a summary of the SCI receiver I/O registers.

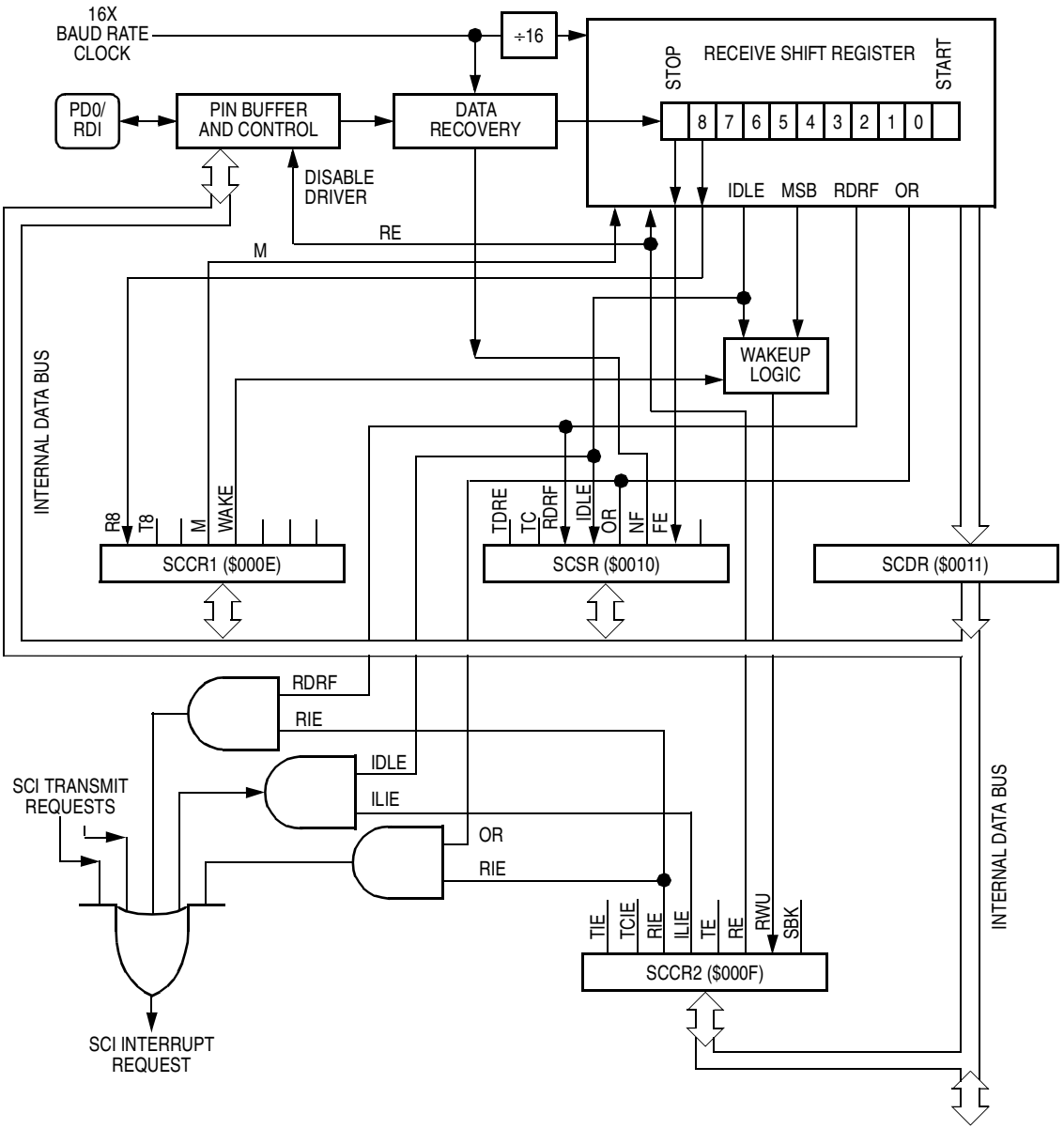


Figure 10-4. SCI Receiver

- Character Length — The receiver can accommodate either 8-bit or 9-bit data. The state of the M bit in SCI control register 1 (SCCR1) determines character length. When receiving 9-bit data, bit R8 in SCCR1 is the ninth bit (bit 8).
- Character Reception — During reception, the receive shift register shifts characters in from the PD0/RDI pin. The SCI data register (SCDR) is the read-only buffer between the internal data bus and the receive shift register.

  After a complete character shifts into the receive shift register, the data portion of the character is transferred to the SCDR, setting the receive data register full (RDRF) flag. The RDRF flag can be used to generate an interrupt.
- Receiver Wakeup — So that the MCU can ignore transmissions intended only for other receivers in multiple-receiver systems, the MCU can be put into a standby state. Setting the receiver wakeup enable (RWU) bit in SCI control register 2 (SCCR2) puts the MCU into a standby state during which receiver interrupts are disabled.

  Either of two conditions on the PD0/RDI pin can bring the MCU out of the standby state:

  – Idle input line condition — If the PD0/RDI pin is at logic 1 long enough for 10 or 11 logic 1s to shift into the receive shift register, receiver interrupts are again enabled.

  – Address mark — If a logic 1 occurs in the most significant bit position of a received character, receiver interrupts are again enabled.

  The state of the WAKE bit in SCCR1 determines which of the two conditions wakes up the MCU.
- Receiver Noise Immunity — The data recovery logic samples each bit 16 times to identify and verify the start bit and to detect noise. Any conflict between noise detection samples sets the noise flag (NF) in the SCSR. The NF bit is set at the same time that the RDRF bit is set.

- Framing Errors — If the data recovery logic does not detect a logic 1 where the stop bit should be in an incoming character, it sets the framing error (FE) bit in the SCSR. The FE bit is set at the same time that the RDRF bit is set.
- Receiver Interrupts — These sources can generate SCI receiver interrupt requests:
  - Receive Data Register Full (RDRF) — The RDRF bit in the SCSR indicates that the receive shift register has transferred a character to the SCDR.
  - Receiver Overrun (OR) — The OR bit in the SCSR indicates that the receive shift register shifted in a new character before the previous character was read from the SCDR.
  - Idle Input (IDLE) — The IDLE bit in the SCSR indicates that 10 or 11 consecutive logic 1s shifted in from the PD0/RDI pin.

10.6 SCI I/O Registers

These I/O registers control and monitor SCI operation:

- SCI data register (SCDR)
- SCI control register 1 (SCCR1)
- SCI control register 2 (SCCR2)
- SCI status register (SCSR)

10.6.1 SCI Data Register

The SCI data register (SCDR) shown in **Figure 10-5** is the buffer for characters received and for characters transmitted.

Address: $0011

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reset:	Unaffected by reset							

Figure 10-5. SCI Data Register (SCDR)

10.6.2 SCI Control Register 1

SCI control register 1 (SCCR1) shown in **Figure 10-6** has these functions:

- Stores ninth SCI data bit received and ninth SCI data bit transmitted
- Controls SCI character length
- Controls SCI wakeup method

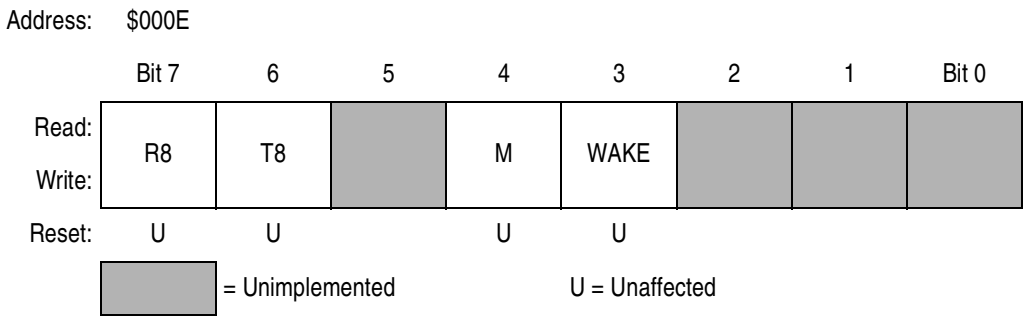


Figure 10-6. SCI Control Register 1 (SCCR1)

R8 — Bit 8 (Received)

When the SCI is receiving 9-bit characters, R8 is the ninth bit of the received character. R8 receives the ninth bit at the same time that the SCDR receives the other eight bits. Reset has no effect on the R8 bit.

T8 — Bit 8 (Transmitted)

When the SCI is transmitting 9-bit characters, T8 is the ninth bit of the transmitted character. T8 is loaded into the transmit shift register at the same time that SCDR is loaded into the transmit shift register. Reset has no effect on the T8 bit.

M — Character Length Bit

This read/write bit determines whether SCI characters are eight or nine bits long. The ninth bit can be used as an extra stop bit, as a receiver wakeup signal, or as a mark or space parity bit. Reset has no effect on the M bit.

1 = 9-bit SCI characters
0 = 8-bit SCI characters

WAKE — Wakeup Bit

This read/write bit determines which condition wakes up the SCI: a logic 1 (address mark) in the most significant bit position of a received character or an idle condition of the PD0/RDI pin. Reset has no effect on the WAKE bit.

1 = Address mark wakeup
0 = Idle line wakeup

10.6.3 SCI Control Register 2

SCI control register 2 (SCCR2) shown in **Figure 10-7** has these functions:

- Enables the SCI receiver and SCI receiver interrupts
- Enables the SCI transmitter and SCI transmitter interrupts
- Enables SCI receiver idle interrupts
- Enables SCI transmission complete interrupts
- Enables SCI wakeup
- Transmits SCI break characters

Address: $000F

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
Reset:	0	0	0	0	0	0	0	0

Figure 10-7. SCI Control Register 2 (SCCR2)

TIE — Transmit Interrupt Enable Bit

This read/write bit enables SCI interrupt requests when the TDRE bit becomes set. Reset clears the TIE bit.

1 = TDRE interrupt requests enabled
0 = TDRE interrupt requests disabled

TCIE — Transmission Complete Interrupt Enable Bit

This read/write bit enables SCI interrupt requests when the TC bit becomes set. Reset clears the TCIE bit.

1 = TC interrupt requests enabled
0 = TC interrupt requests disabled

RIE — Receive Interrupt Enable Bit

This read/write bit enables SCI interrupt requests when the RDRF bit or the OR bit becomes set. Reset clears the RIE bit.

1 = RDRF interrupt requests enabled
0 = RDRF interrupt requests disabled

ILIE — Idle Line Interrupt Enable Bit

This read/write bit enables SCI interrupt requests when the IDLE bit becomes set. Reset clears the ILIE bit.

1 = IDLE interrupt requests enabled
0 = IDLE interrupt requests disabled

TE — Transmit Enable Bit

Setting this read/write bit begins the transmission by sending a preamble of 10 or 11 logic 1s from the transmit shift register to the PD1/TDO pin. Reset clears the TE bit.

1 = Transmission enabled
0 = Transmission disabled

RE — Receive Enable Bit

Setting this read/write bit enables the receiver. Clearing the RE bit disables the receiver and receiver interrupts but does not affect the receiver interrupt flags. Reset clears the RE bit.

1 = Receiver enabled
0 = Receiver disabled

RWU — Receiver Wakeup Enable Bit

This read/write bit puts the receiver in a standby state. Typically, data transmitted to the receiver clears the RWU bit and returns the receiver to normal operation. The WAKE bit in SCCR1 determines whether an

idle input or an address mark brings the receiver out of the standby state. Reset clears the RWU bit.

1 = Standby state
0 = Normal operation

SBK — Send Break Bit

Setting this read/write bit continuously transmits break codes in the form of 10-bit or 11-bit groups of logic 0s. Clearing the SBK bit stops the break codes and transmits a logic 1 as a start bit. Reset clears the SBK bit.

1 = Break codes being transmitted
0 = No break codes being transmitted

10.6.4 SCI Status Register

The SCI status register (SCSR) shown in **Figure 10-8** contains flags to signal these conditions:

- Transfer of SCDR data to transmit shift register complete
- Transmission complete
- Transfer of receive shift register data to SCDR complete
- Receiver input idle
- Receiver overrun
- Noisy data
- Framing error

Address: $0010

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	TDRE	TC	RDRF	IDLE	OR	NF	FE	
Write:								
Reset:	1	1	0	0	0	0	0	U

= Unimplemented U = Unaffected

Figure 10-8. SCI Status Register (SCSR)

TDRE — Transmit Data Register Empty Bit

This clearable, read-only bit is set when the data in the SCDR transfers to the transmit shift register. TDRE generates an interrupt request if the TIE bit in SCCR2 is also set. Clear the TDRE bit by reading the SCSR with TDRE set and then writing to the SCDR. Reset sets the TDRE bit. Software must initialize the TDRE bit to logic 0 to avoid an instant interrupt request when turning on the transmitter.

1 = SCDR data transferred to transmit shift register
0 = SCDR data not transferred to transmit shift register

TC — Transmission Complete Bit

This clearable, read-only bit is set when the TDRE bit is set and no data, preamble, or break character is being transmitted. TC generates an interrupt request if the TCIE bit in SCCR2 is also set. Clear the TC bit by reading the SCSR with TC set and then writing to the SCDR. Reset sets the TC bit. Software must initialize the TC bit to logic 0 to avoid an instant interrupt request when turning on the transmitter.

1 = No transmission in progress
0 = Transmission in progress

RDRF — Receive Data Register Full Bit

This clearable, read-only bit is set when the data in the receive shift register transfers to the SCI data register. RDRF generates an interrupt request if the RIE bit in SCCR2 is also set. Clear the RDRF bit by reading the SCSR with RDRF set and then reading the SCDR. Reset clears the RDRF bit.

1 = Received data available in SCDR
0 = Received data not available in SCDR

IDLE — Receiver Idle Bit

This clearable, read-only bit is set when 10 or 11 consecutive logic 1s appear on the receiver input. IDLE generates an interrupt request if the ILIE bit in SCCR2 is also set. Clear the IDLE bit by reading the SCSR with IDLE set, and then reading the SCDR. Reset clears the IDLE bit.

1 = Receiver input idle
0 = Receiver input not idle

OR — Receiver Overrun Bit

This clearable, read-only bit is set if the SCDR is not read before the receive shift register receives the next word. OR generates an interrupt request if the RIE bit in SCCR2 is also set. The data in the shift register is lost, but the data already in the SCDR is not affected. Clear the OR bit by reading the SCSR with OR set and then reading the SCDR. Reset clears the OR bit.

1 = Receiver shift register full and RDRF = 1
0 = No receiver overrun

NF — Receiver Noise Flag Bit

This clearable, read-only bit is set when noise is detected in data received in the SCI data register. Clear the NF bit by reading the SCSR and then reading the SCDR. Reset clears the NF bit.

1 = Noise detected in SCDR
0 = No noise detected in SCDR

FE — Receiver Framing Error Bit

This clearable, read-only flag is set when a logic 0 is located where a stop bit should be in the character shifted into the receive shift register. If the received word causes both a framing error and an overrun error, the OR bit is set and the FE bit is not set. Clear the FE bit by reading the SCSR and then reading the SCDR. Reset clears the FE bit.

1 = Framing error
0 = No framing error

10.6.5 Baud Rate Register

The baud rate register shown in **Figure 10-9** selects the baud rate for both the receiver and the transmitter.

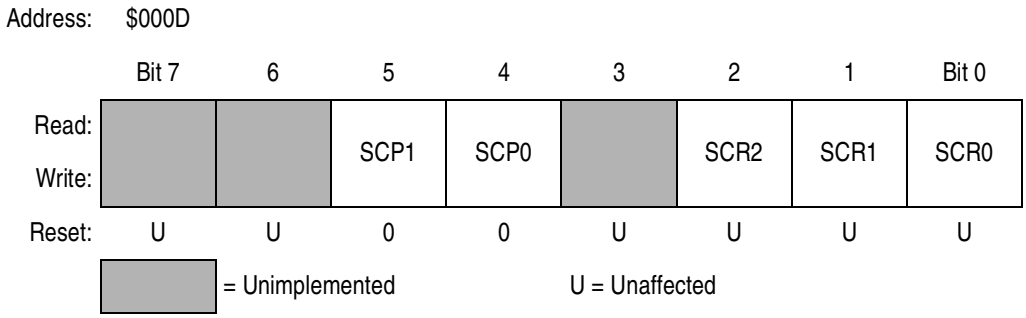


Figure 10-9. Baud Rate Register (Baud)

SCP1 and SCP0 — SCI Prescaler Select Bits

These read/write bits control prescaling of the baud rate generator clock, as shown in **Table 10-1**. Resets clear both SCP1 and SCP0.

Table 10-1. Baud Rate Generator Clock Prescaling

SCP[1:0]	Baud Rate Generator Clock
00	Internal clock ÷ 1
01	Internal clock ÷ 3
10	Internal clock ÷ 4
11	Internal clock ÷ 13

SCR2–SCR0 — SCI Baud Rate Select Bits

These read/write bits select the SCI baud rate, as shown in **Table 10-2**. Reset has no effect on the SCR2–SCR0 bits.

Table 10-2. Baud Rate Selection

SCR[2:1:0]	SCI Baud Rate (Baud)
000	Prescaled clock ÷ 1
001	Prescaled clock ÷ 2
010	Prescaled clock ÷ 4
011	Prescaled clock ÷ 8
100	Prescaled clock ÷ 16
101	Prescaled clock ÷ 32
110	Prescaled clock ÷ 64
111	Prescaled clock ÷ 128

Table 10-3 shows all possible SCI baud rates derived from crystal frequencies of 2 MHz, 4 MHz, and 4.194304 MHz.

Table 10-3. Baud Rate Selection Examples

SCP[1:0]	SCR[2:1:0]	SCI Baud Rate		
		f_{OSC} = 2 MHz	f_{OSC} = 4 MHz	f_{OSC} = 4.194304 MHz
00	000	62.50 Kbaud	125 Kbaud	131.1 Kbaud
00	001	31.25 Kbaud	62.50 Kbaud	65.54 Kbaud
00	010	15.63 Kbaud	31.25 Kbaud	32.77 Kbaud
00	011	7813 baud	15.63 Kbaud	16.38 Kbaud
00	100	3906 baud	7813 baud	8192 baud
00	101	1953 baud	3906 baud	4096 baud
00	110	976.6 baud	1953 baud	2048 baud
00	111	488.3 baud	976.6 baud	1024 baud
01	000	20.83 Kbaud	41.67 Kbaud	43.69 Kbaud
01	001	10.42 Kbaud	20.83 Kbaud	21.85 Kbaud
01	010	5208 baud	10.42 Kbaud	10.92 Kbaud
01	011	2604 baud	5208 baud	5461 baud
01	100	1302 baud	2604 baud	2731 baud
01	101	651.0 baud	1302 baud	1365 baud
01	110	325.5 baud	651.0 baud	682.7 baud
01	111	162.8 baud	325.5 baud	341.3 baud
10	000	15.63 Kbaud	31.25 Kbaud	32.77 Kbaud
10	001	7813 baud	15.63 Kbaud	16.38 Kbaud
10	010	3906 baud	7813 baud	8192 baud
10	011	1953 baud	3906 baud	4906 baud
10	100	976.6 baud	1953 baud	2048 baud
10	101	488.3 baud	976.6 baud	1024 baud
10	110	244.1 baud	488.3 baud	512.0 baud
10	111	122.1 baud	244.1 baud	256.0 baud
11	000	4808 baud	9615 baud	10.08 Kbaud
11	001	2404 baud	4808 baud	5041 baud
11	010	1202 baud	2404 baud	2521 baud
11	011	601.0 baud	1202 baud	1260 baud
11	100	300.5 baud	601.0 baud	630.2 baud
11	101	150.2 baud	300.5 baud	315.1 baud
11	110	75.12 baud	150.2 baud	157.5 baud
11	111	37.56 baud	75.12 baud	78.77 baud

Section 11. Serial Peripheral Interface (SPI)

11.1 Contents

11.2 Introduction

The serial peripheral interface (SPI) module allows full-duplex, synchronous, serial communication with peripheral devices.

11.3 Features

Features of the SPI include:

- Full-duplex operation
- Master and slave modes
- Four programmable master mode frequencies (1.05 MHz maximum)
- 2.1-MHz maximum slave mode frequency
- Serial clock with programmable polarity and phase
- End of transmission interrupt flag
- Write collision error flag
- Bus contention error flag

Figure 11-1 shows the structure of the SPI module. **Figure 11-2** is a summary of the SPI input/output (I/O) registers.

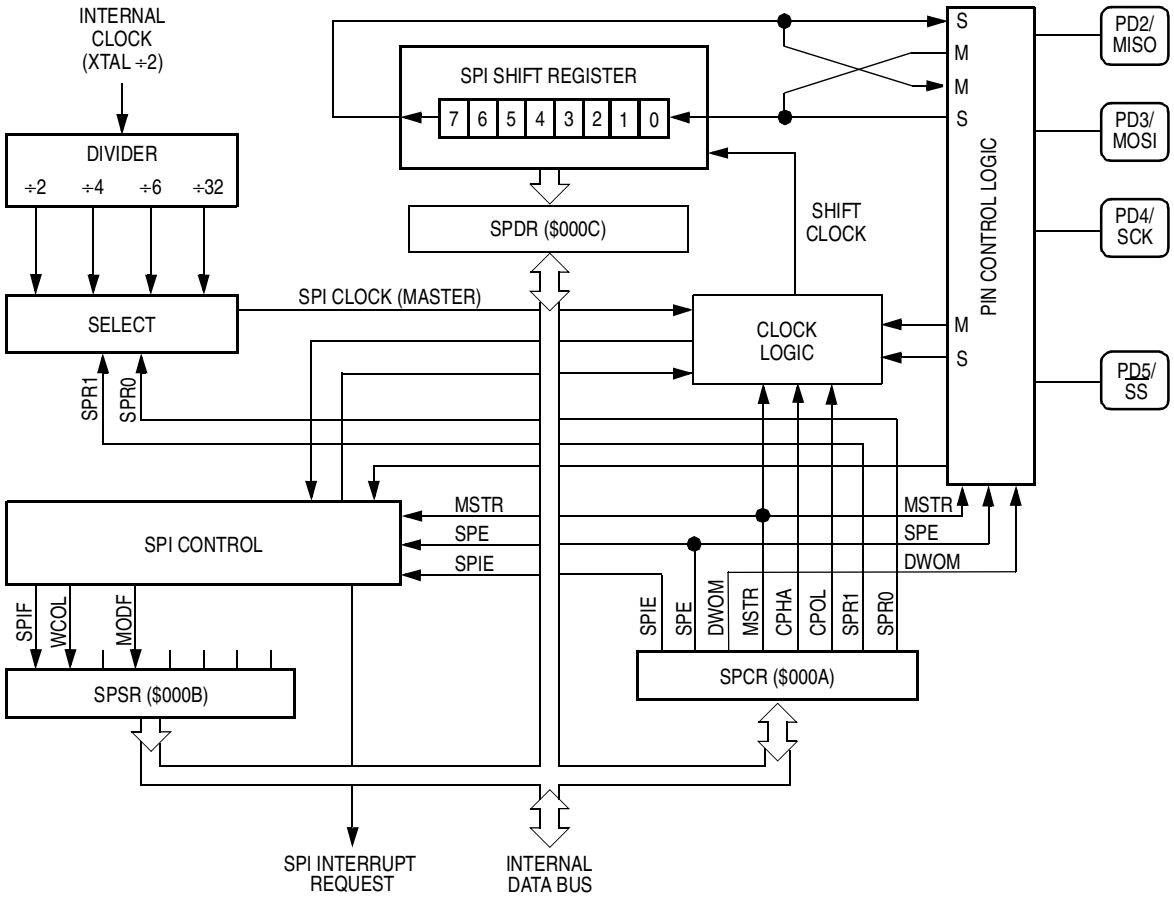


Figure 11-1. SPI Block Diagram

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$000A	SPI Control Register (SPCR) See page 149.	Read:	SPIE	SPE		MSTR	CPOL	CPHA	SPR1	SPR0
		Write:								
		Reset:	0	0		0	U	U	U	U
$000B	SPI Status Register (SPSR) See page 151.	Read:	SPIF	WCOL		MODF				
		Write:								
		Reset:	0	0		0				
$000C	SPI Data Register (SPDR) See page 149.	Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Blt 2	Bit 1	Bit 0
		Write:								
		Reset:	Unaffected by reset							

= Unimplemented U = Unaffected

Figure 11-2. SPI I/O Register Summary

11.4 Operation

The master/slave SPI allows full-duplex, synchronous, serial communication between the microcontroller unit (MCU) and peripheral devices, including other MCUs. As the 8-bit shift register of a master SPI transmits each byte to another device, a byte from the receiving device enters the master SPI shift register. A clock signal from the master SPI synchronizes data transmission.

Only a master SPI can initiate transmissions. Software begins the transmission from a master SPI by writing to the SPI data register (SPDR). The SPDR does not buffer data being transmitted from the SPI. Data written to the SPDR goes directly into the shift register and begins the transmission immediately under the control of the serial clock. The transmission ends after eight cycles of the serial clock when the SPI flag (SPIF) becomes set. At the same time that SPIF becomes set, the data shifted into the master SPI from the receiving device transfers to the SPDR. The SPDR buffers data being received by the SPI. Before the master SPI sends the next byte, software must clear the SPIF bit by reading the SPSR and then accessing the SPDR.

In a slave SPI, data enters the shift register under the control of the serial clock from the master SPI. After a byte enters the shift register of a slave SPI, it transfers to the SPDR. To prevent an overrun condition, slave software must then read the byte in the SPDR before another byte enters the shift register and is ready to transfer to the SPDR.

Figure 11-3 shows how a master SPI exchanges data with a slave SPI.

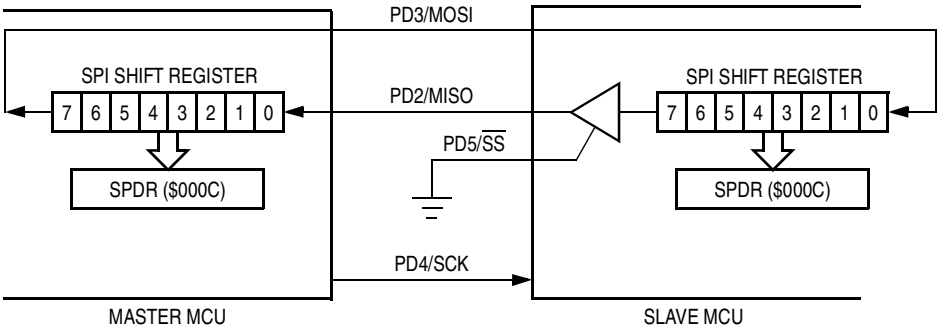


Figure 11-3. Master/Slave Connections

11.4.1 Pin Functions in Master Mode

Setting the MSTR bit in the SPI control register (SPCR) configures the SPI for operation in master mode. The master-mode functions of the SPI pins are:

- PD4/SCK (serial clock) — In master mode, the PD4/SCK pin is the synchronizing clock output.
- PD3/MOSI (master output, slave input) — In master mode, the PD3/MOSI pin is the serial output.
- PD2/MISO (master input, slave output) — In master mode, the PD2/MISO pin is configured as the serial input.
- PD5/$\overline{SS}$ (slave select) — In master mode, the PD5/$\overline{SS}$ pin protects against driver contention caused by the simultaneous operation of two SPIs in master mode. A logic 0 on the PD5/$\overline{SS}$ pin of a master SPI disables the SPI, clears the MSTR bit, and sets the mode-fault flag (MODF).

11.4.2 Pin Functions in Slave Mode

Clearing the MSTR bit in the SPCR configures the SPI for operation in slave mode. The slave-mode functions of the SPI pins are:

- PD4/SCK (serial clock) — In slave mode, the PD4/SCK pin is the input for the synchronizing clock signal from the master SPI.
- PD3/MOSI (master output, slave input) — In slave mode, the PD3/MOSI pin is the serial input.
- PD2/MISO (master input, slave output) — In slave mode, the PD2/MISO pin is the serial output.
- PD5/$\overline{SS}$ (slave select) — In slave mode, the PD5/$\overline{SS}$ pin enables the SPI for data and serial clock reception from a master SPI.

When CPHA = 0, the shift clock is the OR of $\overline{SS}$ with SCK. In this clock phase mode, SS must go high between successive characters in an SPI message. When CPHA = 1, $\overline{SS}$ may be left low for several SPI characters. In cases with only one SPI slave MCU, the slave MCU $\overline{SS}$ line can be tied to V_{SS} as long as CPHA = 1 clock modes are used.

The WCOL flag bit can be improperly set when attempting the first transmission after a reset if these conditions are present: MSTR = 0, CPOL = 0, CPHA = 1, $\overline{SS}$ pin = 0, and SCK pin = 1. The reset states of the CPOL and CPHA bits are 0 and 1, respectively. Under normal operating conditions (CPOL = 0, CPHA = 1), the SCK input will be low.

The incorrect setting of the WCOL bit can be prevented in two ways:

1. Send a dummy transmission after reset, clear the WCOL flag, and then proceed with the real transmission.
2. Use the MSTR bit in the SPCR (SPI control register). This is accomplished by setting the MSTR bit at the same time the CPOL and CPHA bits are programmed to the desired logic levels. Then, the data register can be written to if desired. After this, the MSTR bit should be set to a logic 0, the SPE (SPI enable bit) should be set to a logic 1, and the CPOL, CPHA, SPR1, and SPR0 bits set to the desired logic levels. If this procedure is followed after a reset and before the first access to the SPDR, the WCOL flag will not be set.

Example:

```
LDA #$1C   ; MSTR = 1, CPOL = 1, CPHA = 1,
           ; SPR1 = SPR0 = 0
STA SPCR   ; SPI control register
LDA #$4C   ; MSTR = 0, SPE = 1, CPOL = 1, CPHA = 1,
           ; SPR1 = SPR0 = 0
STA SPCR   ; SPI control register
```

11.5 Multiple-SPI Systems

In a multiple-SPI system, all PD4/SCK pins are connected together, all PD3/MOSI pins are connected together, and all PD2/MISO pins are connected together.

Before a transmission, one SPI is configured as master and the rest are configured as slaves. **Figure 11-4** is a block diagram showing a single master SPI and three slave SPIs.

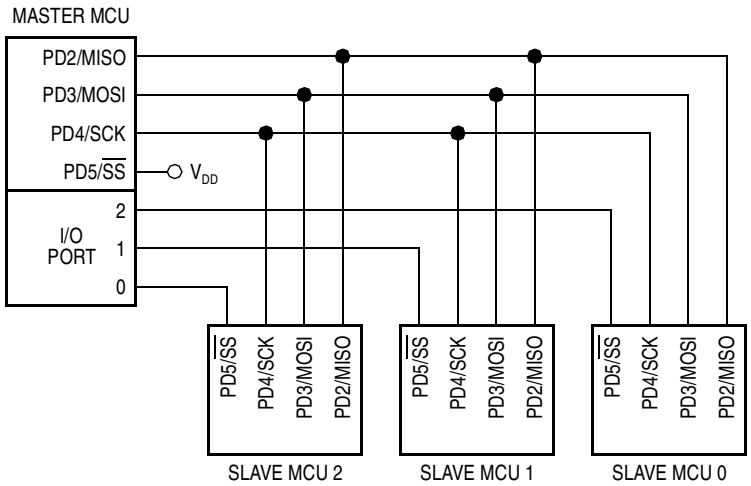


Figure 11-4. One Master and Three Slaves Block Diagram

Figure 11-5 is another block diagram with two master/slave SPIs and three slave SPIs.

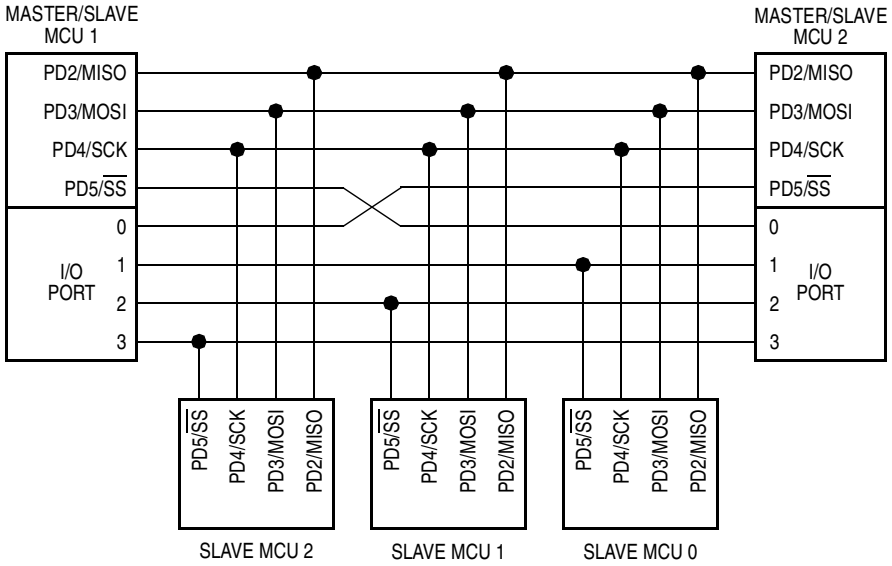


Figure 11-5. Two Master/Slaves and Three Slaves Block Diagram

11.6 Serial Clock Polarity and Phase

To accommodate the different serial communication requirements of peripheral devices, software can change the phase and polarity of the SPI serial clock. The clock polarity bit (CPOL) and the clock phase bit (CPHA), both in the SPCR, control the timing relationship between the serial clock and the transmitted data. **Figure 11-6** shows how the CPOL and CPHA bits affect the clock/data timing.

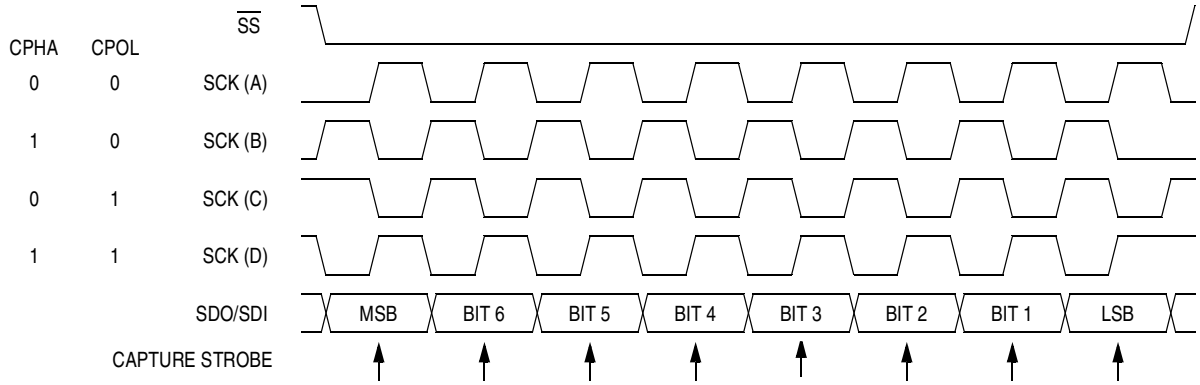


Figure 11-6. SPI Clock/Data Timing

11.7 SPI Error Conditions

These conditions produce SPI system errors:

- Bus contention caused by multiple master SPIs (mode fault error)
- Writing to the SPDR during a transmission (write-collision error)
- Failing to read the SPDR before the next incoming byte sets the SPIF bit (overrun error)

11.7.1 Mode Fault Error

A mode fault error results when a logic 0 occurs on the PD5/$\overline{SS}$ pin of a master SPI. The MCU takes these actions when a mode fault error occurs:

- Puts the SPI in slave mode by clearing the MSTR bit
- Disables the SPI by clearing the SPE bit
- Sets the MODF bit

11.7.2 Write Collision Error

Writing to the SPDR during a transmission causes a write collision error and sets the WCOL bit in the SPSR. Either a master SPI or a slave SPI can generate a write collision error.

- Master — A master SPI can cause a write collision error by writing to the SPDR while the previously written byte is still being shifted out to the PD3/MOSI pin. The error does not affect the transmission of the previously written byte, but the byte that caused the error is lost.
- Slave — A slave SPI can cause a write collision error in either of two ways, depending on the state of the CPHA bit:
  - CPHA = 0 — A slave SPI can cause a write collision error by writing to the SPDR while the PD5/$\overline{SS}$ pin is at logic 0. The error does not affect the byte in the SPDR, but the byte that caused the error is lost.

– CPHA = 1 — A slave SPI can cause a write collision error by writing to the SPDR while receiving a transmission, that is, between the first active SCK edge and the end of the eighth SCK cycle. The error does not affect the transmission from the master SPI, but the byte that caused the error is lost.

11.7.3 Overrun Error

Failing to read the byte in the SPDR before a subsequent byte enters the shift register causes an overrun condition. In an overrun condition, all incoming data is lost until software clears SPIF. The overrun condition has no flag.

11.8 SPI Interrupts

The SPIF bit in the SPSR indicates a byte has shifted into or out of the SPDR. The SPIF bit is a source of SPI interrupt requests. The SPI interrupt enable bit (SPIE) in the SPCR is the local mask for SPIF interrupts.

The MODF bit in the SPSR indicates a mode error and is a source of SPI interrupt requests. The MODF bit is set when a logic 0 occurs on the PD5/$\overline{SS}$ pin while the MSTR bit is set. The SPI interrupt enable bit (SPIE) in the SPCR is the local mask for MODF interrupts.

11.9 SPI I/O Registers

These input/output (I/O) registers control and monitor SPI operation:

- SPI data register (SPDR)
- SPI control register (SPCR)
- SPI status register (SPSR)

11.9.1 SPI Data Register

The SPDR shown in **Figure 11-7** is the read buffer for characters received by the SPI. Writing a byte to the SPDR places the byte directly into the SPI shift register.

Address: $000C

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reset:	Unaffected by reset							

Figure 11-7. SPI Data Register (SPDR)

11.9.2 SPI Control Register

- Enables SPI interrupt requests
- Enables the SPI
- Configures the SPI as master or slave
- Selects serial clock polarity, phase, and frequency

Address: $000A

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	SPIE	SPE		MSTR	CPOL	CPHA	SPR1	SPR0
Reset:	0	0		0	U	U	U	U

= Unimplemented U = Unaffected

Figure 11-8. SPI Control Register (SPCR)

SPIE — SPI Interrupt Enable Bit

This read/write bit enables SPI interrupts. Reset clears the SPIE bit.

1 = SPI interrupts enabled
0 = SPI interrupts disabled

SPI — SPI Enable Bit

This read/write bit enables the SPI. Reset clears the SPE bit.

1 = SPI enabled
0 = SPI disabled

MSTR — Master Bit

This read/write bit selects master mode operation or slave mode operation. Reset clears the MSTR bit.

1 = Master mode
0 = Slave mode

CPOL — Clock Polarity Bit

This read/write bit determines the logic state of the PD4/SCK pin between transmissions. To transmit data between SPIs, the SPIs must have identical CPOL bits. Reset has no effect on the CPOL bit.

1 = PD4/SCK pin at logic 1 between transmissions
0 = PD4/SCK pin at logic 0 between transmissions

CPHA — Clock Phase Bit

This read/write bit controls the timing relationship between the serial clock and SPI data. To transmit data between SPIs, the SPIs must have identical CPHA bits. When CPHA = 0, the PD5/$\overline{SS}$ pin of the slave SPI must be set to logic 1 between bytes. Reset has no effect on the CPHA bit.

1 = Edge following first active edge on PD4/SCK latches data
0 = First active edge on PD4/SCK latches data

SPR1 and SPR0 — SPI Clock Rate Bits

These read/write bits select the master mode serial clock rate, as shown in **Table 11-1**. The SPR1 and SPR0 bits of a slave SPI have no effect on the serial clock. Reset has no effect on SPR1 and SPR0.

Table 11-1. SPI Clock Rate Selection

SPR[1:0]	SPI Clock Rate
00	Internal Clock ÷ 2
01	Internal Clock ÷ 4
10	Internal Clock ÷ 16
11	Internal Clock ÷ 32

11.9.3 SPI Status Register

The SPSR shown in **Figure 11-9** contains flags to signal these conditions:

- SPI transmission complete
- Write collision
- Mode fault

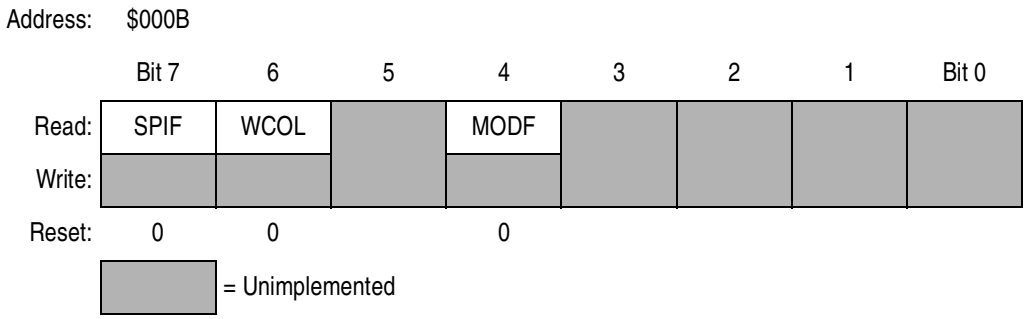


Figure 11-9. SPI Status Register (SPSR)

SPIF — SPI Flag

This clearable, read-only bit is set each time a byte shifts out of or into the shift register. SPIF generates an interrupt request if the SPIE bit in the SPCR is also set. Clear SPIF by reading the SPSR with SPIF set and then reading or writing the SPDR. Reset clears the SPIF bit.

1 = Transmission complete
0 = Transmission not complete

WCOL — Write Collision Bit

This clearable, read-only flag is set when software writes to the SPDR while a transmission is in progress. Clear the WCOL bit by reading the SPSR with WCOL set and then reading or writing the SPDR. Reset clears WCOL.

1 = Invalid write to SPDR
0 = No invalid write to SPDR

MODF — Mode Fault Bit

This clearable, read-only bit is set when a logic 0 occurs on the PD5/$\overline{SS}$ pin while the MSTR bit is set. MODF generates an interrupt request if the SPIE bit is also set. Clear the MODF bit by reading the SPSR with MODF set and then writing to the SPCR. Reset clears MODF.

1 = PD5/$\overline{SS}$ pulled low while MSTR bit set
0 = PD5/$\overline{SS}$ not pulled low while MSTR bit set

Section 12. Instruction Set

12.1 Contents

12.2 Introduction

The MCU instruction set has 62 instructions and uses eight addressing modes. The instructions include all those of the M146805 CMOS Family plus one more: the unsigned multiply (MUL) instruction. The MUL instruction allows unsigned multiplication of the contents of the accumulator (A) and the index register (X). The high-order product is stored in the index register, and the low-order product is stored in the accumulator.

12.3 Addressing Modes

The CPU uses eight addressing modes for flexibility in accessing data. The addressing modes provide eight different ways for the CPU to find the data required to execute an instruction.

The eight addressing modes are:

- Inherent
- Immediate
- Direct
- Extended
- Indexed, no offset
- Indexed, 8-bit offset
- Indexed, 16-bit offset
- Relative

12.3.1 Inherent

Inherent instructions are those that have no operand, such as return from interrupt (RTI) and stop (STOP). Some of the inherent instructions act on data in the CPU registers, such as set carry flag (SEC) and increment accumulator (INCA). Inherent instructions require no operand address and are one byte long.

12.3.2 Immediate

Immediate instructions are those that contain a value to be used in an operation with the value in the accumulator or index register. Immediate instructions require no operand address and are two bytes long. The opcode is the first byte, and the immediate data value is the second byte.

12.3.3 Direct

Direct instructions can access any of the first 256 memory locations with two bytes. The first byte is the opcode, and the second is the low byte of the operand address. In direct addressing, the CPU automatically uses $00 as the high byte of the operand address.

12.3.4 Extended

Extended instructions use three bytes and can access any address in memory. The first byte is the opcode; the second and third bytes are the high and low bytes of the operand address.

When using the Motorola assembler, the programmer does not need to specify whether an instruction is direct or extended. The assembler automatically selects the shortest form of the instruction.

12.3.5 Indexed, No Offset

Indexed instructions with no offset are 1-byte instructions that can access data with variable addresses within the first 256 memory locations. The index register contains the low byte of the effective address of the operand. The CPU automatically uses $00 as the high byte, so these instructions can address locations $0000–$00FF.

Indexed, no offset instructions are often used to move a pointer through a table or to hold the address of a frequently used RAM or I/O location.

12.3.6 Indexed, 8-Bit Offset

Indexed, 8-bit offset instructions are 2-byte instructions that can access data with variable addresses within the first 511 memory locations. The CPU adds the unsigned byte in the index register to the unsigned byte following the opcode. The sum is the effective address of the operand. These instructions can access locations $0000–$01FE.

Indexed 8-bit offset instructions are useful for selecting the kth element in an n-element table. The table can begin anywhere within the first 256 memory locations and could extend as far as location 510 ($01FE). The k value is typically in the index register, and the address of the beginning of the table is in the byte following the opcode.

12.3.7 Indexed, 16-Bit Offset

Indexed, 16-bit offset instructions are 3-byte instructions that can access data with variable addresses at any location in memory. The CPU adds the unsigned byte in the index register to the two unsigned bytes following the opcode. The sum is the effective address of the operand. The first byte after the opcode is the high byte of the 16-bit offset; the second byte is the low byte of the offset.

Indexed, 16-bit offset instructions are useful for selecting the kth element in an n-element table anywhere in memory.

As with direct and extended addressing, the Motorola assembler determines the shortest form of indexed addressing.

12.3.8 Relative

Relative addressing is only for branch instructions. If the branch condition is true, the CPU finds the effective branch destination by adding the signed byte following the opcode to the contents of the program counter. If the branch condition is not true, the CPU goes to the next instruction. The offset is a signed, two's complement byte that gives a branching range of –128 to +127 bytes from the address of the next location after the branch instruction.

When using the Motorola assembler, the programmer does not need to calculate the offset, because the assembler determines the proper offset and verifies that it is within the span of the branch.

12.4 Instruction Types

The MCU instructions fall into five categories:

- Register/memory instructions
- Read-modify-write instructions
- Jump/branch instructions
- Bit manipulation instructions
- Control instructions

12.4.1 Register/Memory Instructions

These instructions operate on CPU registers and memory locations. Most of them use two operands. One operand is in either the accumulator or the index register. The CPU finds the other operand in memory.

Table 12-1. Register/Memory Instructions

Instruction	Mnemonic
Add memory byte and carry bit to accumulator	ADC
Add memory byte to accumulator	ADD
AND memory byte with accumulator	AND
Bit test accumulator	BIT
Compare accumulator	CMP
Compare index register with memory byte	CPX
Exclusive OR accumulator with memory byte	EOR
Load accumulator with memory byte	LDA
Load Index register with memory byte	LDX
Multiply	MUL
OR accumulator with memory byte	ORA
Subtract memory byte and carry bit from accumulator	SBC
Store accumulator in memory	STA
Store index register in memory	STX
Subtract memory byte from accumulator	SUB

12.4.2 Read-Modify-Write Instructions

These instructions read a memory location or a register, modify its contents, and write the modified value back to the memory location or to the register.

NOTE: *Do not use read-modify-write operations on write-only registers.*

Table 12-2. Read-Modify-Write Instructions

Instruction	Mnemonic
Arithmetic shift left (same as LSL)	ASL
Arithmetic shift right	ASR
Bit clear	BCLR(1)
Bit set	BSET(1)
Clear register	CLR
Complement (one's complement)	COM
Decrement	DEC
Increment	INC
Logical shift left (same as ASL)	LSL
Logical shift right	LSR
Negate (two's complement)	NEG
Rotate left through carry bit	ROL
Rotate right through carry bit	ROR
Test for negative or zero	TST(2)

1. Unlike other read-modify-write instructions, BCLR and BSET use only direct addressing.
2. TST is an exception to the read-modify-write sequence because it does not write a replacement value.

12.4.3 Jump/Branch Instructions

Jump instructions allow the CPU to interrupt the normal sequence of the program counter. The unconditional jump instruction (JMP) and the jump-to-subroutine instruction (JSR) have no register operand. Branch instructions allow the CPU to interrupt the normal sequence of the program counter when a test condition is met. If the test condition is not met, the branch is not performed.

The BRCLR and BRSET instructions cause a branch based on the state of any readable bit in the first 256 memory locations. These 3-byte instructions use a combination of direct addressing and relative addressing. The direct address of the byte to be tested is in the byte following the opcode. The third byte is the signed offset byte. The CPU finds the effective branch destination by adding the third byte to the program counter if the specified bit tests true. The bit to be tested and its condition (set or clear) is part of the opcode. The span of branching is from –128 to +127 from the address of the next location after the branch instruction. The CPU also transfers the tested bit to the carry/borrow bit of the condition code register.

Table 12-3. Jump and Branch Instructions

Instruction	Mnemonic
Branch if carry bit clear	BCC
Branch if carry bit set	BCS
Branch if equal	BEQ
Branch if half-carry bit clear	BHCC
Branch if half-carry bit set	BHCS
Branch if higher	BHI
Branch if higher or same	BHS
Branch if $\overline{\text{IRQ}}$ pin high	BIH
Branch if $\overline{\text{IRQ}}$ pin low	BIL
Branch if lower	BLO
Branch if lower or same	BLS
Branch if interrupt mask clear	BMC
Branch if minus	BMI
Branch if interrupt mask set	BMS
Branch if not equal	BNE
Branch if plus	BPL
Branch always	BRA
Branch if bit clear	BRCLR
Branch never	BRN
Branch if bit set	BRSET
Branch to subroutine	BSR
Unconditional jump	JMP
Jump to subroutine	JSR

12.4.4 Bit Manipulation Instructions

The CPU can set or clear any writable bit in the first 256 bytes of memory, which includes I/O registers and on-chip RAM locations. The CPU can also test and branch based on the state of any bit in any of the first 256 memory locations.

Table 12-4. Bit Manipulation Instructions

Instruction	Mnemonic
Bit clear	BCLR
Branch if bit clear	BRCLR
Branch if bit set	BRSET
Bit set	BSET

12.4.5 Control Instructions

These instructions act on CPU registers and control CPU operation during program execution.

Table 12-5. Control Instructions

Instruction	Mnemonic
Clear carry bit	CLC
Clear interrupt mask	CLI
No operation	NOP
Reset stack pointer	RSP
Return from interrupt	RTI
Return from subroutine	RTS
Set carry bit	SEC
Set interrupt mask	SEI
Stop oscillator and enable $\overline{\text{IRQ}}$ pin	STOP
Software interrupt	SWI
Transfer accumulator to index register	TAX
Transfer index register to accumulator	TXA
Stop CPU clock and enable interrupts	WAIT

12.5 Instruction Set Summary

Table 12-6. Instruction Set Summary (Sheet 1 of 6)

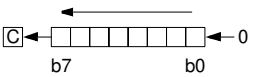
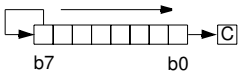
Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
ADC #*opr* ADC *opr* ADC *opr* ADC *opr*,X ADC *opr*,X ADC ,X	Add with Carry	A ← (A) + (M) + (C)	↕	—	↕	↕	↕	IMM DIR EXT IX2 IX1 IX	A9 B9 C9 D9 E9 F9	ii dd hh ll ee ff ff	2 3 4 5 4 3
ADD #*opr* ADD *opr* ADD *opr* ADD *opr*,X ADD *opr*,X ADD ,X	Add without Carry	A ← (A) + (M)	↕	—	↕	↕	↕	IMM DIR EXT IX2 IX1 IX	AB BB CB DB EB FB	ii dd hh ll ee ff ff	2 3 4 5 4 3
AND #*opr* AND *opr* AN*D opr* AND *opr*,X AND *opr*,X AND ,X	Logical AND	A ← (A) ∧ (M)	—	—	↕	↕	—	IMM DIR EXT IX2 IX1 IX	A4 B4 C4 D4 E4 F4	ii dd hh ll ee ff ff	2 3 4 5 4 3
ASL *opr* ASLA ASLX ASL *opr*,X ASL ,X	Arithmetic Shift Left (Same as LSL)		—	—	↕	↕	↕	DIR INH INH IX1 IX	38 48 58 68 78	dd ff 	5 3 3 6 5
ASR *opr* ASRA ASRX ASR *opr*,X ASR ,X	Arithmetic Shift Right		—	—	↕	↕	↕	DIR INH INH IX1 IX	37 47 57 67 77	dd ff 	5 3 3 6 5
BCC *rel*	Branch if Carry Bit Clear	PC ← (PC) + 2 + *rel* ? C = 0	—	—	—	—	—	REL	24	rr	3
BCLR *n opr*	Clear Bit n	Mn ← 0	—	—	—	—	—	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	11 13 15 17 19 1B 1D 1F	dd dd dd dd dd dd dd dd	5 5 5 5 5 5 5 5
BCS *rel*	Branch if Carry Bit Set (Same as BLO)	PC ← (PC) + 2 + *rel* ? C = 1	—	—	—	—	—	REL	25	rr	3
BEQ *rel*	Branch if Equal	PC ← (PC) + 2 + *rel* ? Z = 1	—	—	—	—	—	REL	27	rr	3
BHCC *rel*	Branch if Half-Carry Bit Clear	PC ← (PC) + 2 + *rel* ? H = 0	—	—	—	—	—	REL	28	rr	3
BHCS *rel*	Branch if Half-Carry Bit Set	PC ← (PC) + 2 + *rel* ? H = 1	—	—	—	—	—	REL	29	rr	3
BHI *rel*	Branch if Higher	PC ← (PC) + 2 + *rel* ? C ∨ Z = 0	—	—	—	—	—	REL	22	rr	3
BHS *rel*	Branch if Higher or Same	PC ← (PC) + 2 + *rel* ? C = 0	—	—	—	—	—	REL	24	rr	3

Table 12-6. Instruction Set Summary (Sheet 2 of 6)

Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
BIH *rel*	Branch if IRQ Pin High	PC ← (PC) + 2 + *rel* ? IRQ = 1	—	—	—	—	—	REL	2F	rr	3
BIL *rel*	Branch if IRQ Pin Low	PC ← (PC) + 2 + *rel* ? IRQ = 0	—	—	—	—	—	REL	2E	rr	3
BIT #*opr* BIT *opr* BIT *opr* BIT *opr*,X BIT *opr*,X BIT ,X	Bit Test Accumulator with Memory Byte	(A) ∧ (M)	—	—	↕	↕	—	IMM DIR EXT IX2 IX1 IX	A5 B5 C5 D5 E5 F5	ii dd hh ll ee ff ff	2 3 4 5 4 3
BLO *rel*	Branch if Lower (Same as BCS)	PC ← (PC) + 2 + *rel* ? C = 1	—	—	—	—	—	REL	25	rr	3
BLS *rel*	Branch if Lower or Same	PC ← (PC) + 2 + *rel* ? C ∨ Z = 1	—	—	—	—	—	REL	23	rr	3
BMC *rel*	Branch if Interrupt Mask Clear	PC ← (PC) + 2 + *rel* ? I = 0	—	—	—	—	—	REL	2C	rr	3
BMI *rel*	Branch if Minus	PC ← (PC) + 2 + *rel* ? N = 1	—	—	—	—	—	REL	2B	rr	3
BMS *rel*	Branch if Interrupt Mask Set	PC ← (PC) + 2 + *rel* ? I = 1	—	—	—	—	—	REL	2D	rr	3
BNE *rel*	Branch if Not Equal	PC ← (PC) + 2 + *rel* ? Z = 0	—	—	—	—	—	REL	26	rr	3
BPL *rel*	Branch if Plus	PC ← (PC) + 2 + *rel* ? N = 0	—	—	—	—	—	REL	2A	rr	3
BRA *rel*	Branch Always	PC ← (PC) + 2 + *rel* ? 1 = 1	—	—	—	—	—	REL	20	rr	3
BRCLR *n opr rel*	Branch if Bit n Clear	PC ← (PC) + 2 + *rel* ? Mn = 0	—	—	—	—	↕	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	01 03 05 07 09 0B 0D 0F	dd rr dd rr dd rr dd rr dd rr dd rr dd rr dd rr	5 5 5 5 5 5 5 5
BRN *rel*	Branch Never	PC ← (PC) + 2 + *rel* ? 1 = 0	—	—	—	—	—	REL	21	rr	3
BRSET *n opr rel*	Branch if Bit n Set	PC ← (PC) + 2 + *rel* ? Mn = 1	—	—	—	—	↕	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	00 02 04 06 08 0A 0C 0E	dd rr dd rr dd rr dd rr dd rr dd rr dd rr dd rr	5 5 5 5 5 5 5 5
BSET *n opr*	Set Bit n	Mn ← 1	—	—	—	—	—	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	10 12 14 16 18 1A 1C 1E	dd dd dd dd dd dd dd dd	5 5 5 5 5 5 5 5
BSR *rel*	Branch to Subroutine	PC ← (PC) + 2; push (PCL) SP ← (SP) – 1; push (PCH) SP ← (SP) – 1 PC ← (PC) + *rel*	—	—	—	—	—	REL	AD	rr	6
CLC	Clear Carry Bit	C ← 0	—	—	—	—	0	INH	98		2
CLI	Clear Interrupt Mask	I ← 0	—	0	—	—	—	INH	9A		2

Table 12-6. Instruction Set Summary (Sheet 3 of 6)

Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
CLR *opr* CLRA CLRX CLR *opr*,X CLR ,X	Clear Byte	M ← $00 A ← $00 X ← $00 M ← $00 M ← $00	—	—	0	1	—	DIR INH INH IX1 IX	3F 4F 5F 6F 7F	dd ff 	5 3 3 6 5
CMP #*opr* CMP *opr* CMP *opr* CMP *opr*,X CMP *opr*,X CMP ,X	Compare Accumulator with Memory Byte	(A) – (M)	—	—	↕	↕	↕	IMM DIR EXT IX2 IX1 IX	A1 B1 C1 D1 E1 F1	ii dd hh ll ee ff ff 	2 3 4 5 4 3
COM *opr* COMA COMX COM *opr*,X COM ,X	Complement Byte (One's Complement)	M ← ($\overline{M}$) = $FF – (M) A ← ($\overline{A}$) = $FF – (A) X ← ($\overline{X}$) = $FF – (X) M ← ($\overline{M}$) = $FF – (M) M ← ($\overline{M}$) = $FF – (M)	—	—	↕	↕	1	DIR INH INH IX1 IX	33 43 53 63 73	dd ff 	5 3 3 6 5
CPX #*opr* CPX *opr* CPX *opr* CPX *opr*,X CPX *opr*,X CPX ,X	Compare Index Register with Memory Byte	(X) – (M)	—	—	↕	↕	↕	IMM DIR EXT IX2 IX1 IX	A3 B3 C3 D3 E3 F3	ii dd hh ll ee ff ff 	2 3 4 5 4 3
DEC *opr* DECA DECX DEC *opr*,X DEC ,X	Decrement Byte	M ← (M) – 1 A ← (A) – 1 X ← (X) – 1 M ← (M) – 1 M ← (M) – 1	—	—	↕	↕	—	DIR INH INH IX1 IX	3A 4A 5A 6A 7A	dd ff 	5 3 3 6 5
EOR #*opr* EOR *opr* EOR *opr* EOR *opr*,X EOR *opr*,X EOR ,X	EXCLUSIVE OR Accumulator with Memory Byte	A ← (A) ⊕ (M)	—	—	↕	↕	—	IMM DIR EXT IX2 IX1 IX	A8 B8 C8 D8 E8 F8	ii dd hh ll ee ff ff 	2 3 4 5 4 3
INC *opr* INCA INCX INC *opr*,X INC ,X	Increment Byte	M ← (M) + 1 A ← (A) + 1 X ← (X) + 1 M ← (M) + 1 M ← (M) + 1	—	—	↕	↕	—	DIR INH INH IX1 IX	3C 4C 5C 6C 7C	dd ff 	5 3 3 6 5
JMP *opr* JMP *opr* JMP *opr*,X JMP *opr*,X JMP ,X	Unconditional Jump	PC ← Jump Address	—	—	—	—	—	DIR EXT IX2 IX1 IX	BC CC DC EC FC	dd hh ll ee ff ff 	2 3 4 3 2

Table 12-6. Instruction Set Summary (Sheet 4 of 6)

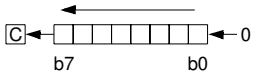
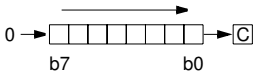
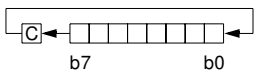
Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
JSR *opr* JSR *opr* JSR *opr*,X JSR *opr*,X JSR ,X	Jump to Subroutine	PC ← (PC) + n (n = 1, 2, or 3) Push (PCL); SP ← (SP) – 1 Push (PCH); SP ← (SP) – 1 PC ← Effective Address	—	—	—	—	—	DIR EXT IX2 IX1 IX	BD CD DD ED FD	dd hh ll ee ff ff	5 6 7 6 5
LDA #*opr* LDA *opr* LDA *opr* LDA *opr*,X LDA *opr*,X LDA ,X	Load Accumulator with Memory Byte	A ← (M)	—	—	↕	↕	—	IMM DIR EXT IX2 IX1 IX	A6 B6 C6 D6 E6 F6	ii dd hh ll ee ff ff	2 3 4 5 4 3
LDX #*opr* LDX *opr* LDX *opr* LDX *opr*,X LDX *opr*,X LDX ,X	Load Index Register with Memory Byte	X ← (M)	—	—	↕	↕	—	IMM DIR EXT IX2 IX1 IX	AE BE CE DE EE FE	ii dd hh ll ee ff ff	2 3 4 5 4 3
LSL *opr* LSLA LSLX LSL *opr*,X LSL ,X	Logical Shift Left (Same as ASL)		—	—	↕	↕	↕	DIR INH INH IX1 IX	38 48 58 68 78	dd ff 	5 3 3 6 5
LSR *opr* LSRA LSRX LSR *opr*,X LSR ,X	Logical Shift Right		—	—	0	↕	↕	DIR INH INH IX1 IX	34 44 54 64 74	dd ff 	5 3 3 6 5
MUL	Unsigned Multiply	X : A ← (X) × (A)	0	—	—	—	0	INH	42		1 1
NEG *opr* NEGA NEGX NEG *opr*,X NEG ,X	Negate Byte (Two's Complement)	M ← –(M) = $00 – (M) A ← –(A) = $00 – (A) X ← –(X) = $00 – (X) M ← –(M) = $00 – (M) M ← –(M) = $00 – (M)	—	—	↕	↕	↕	DIR INH INH IX1 IX	30 40 50 60 70	dd ff 	5 3 3 6 5
NOP	No Operation		—	—	—	—	—	INH	9D		2
ORA #*opr* ORA *opr* ORA *opr* ORA *opr*,X ORA *opr*,X ORA ,X	Logical OR Accumulator with Memory	A ← (A) ∨ (M)	—	—	↕	↕	—	IMM DIR EXT IX2 IX1 IX	AA BA CA DA EA FA	ii dd hh ll ee ff ff	2 3 4 5 4 3
ROL *opr* ROLA ROLX ROL *opr*,X ROL ,X	Rotate Byte Left through Carry Bit		—	—	↕	↕	↕	DIR INH INH IX1 IX	39 49 59 69 79	dd ff 	5 3 3 6 5

Table 12-6. Instruction Set Summary (Sheet 5 of 6)

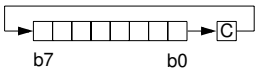
Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
ROR *opr* RORA RORX ROR *opr*,X ROR ,X	Rotate Byte Right through Carry Bit		—	—	↕	↕	↕	DIR INH INH IX1 IX	36 46 56 66 76	dd ff 	5 3 3 6 5
RSP	Reset Stack Pointer	SP ← $00FF	—	—	—	—	—	INH	9C		2
RTI	Return from Interrupt	SP ← (SP) + 1; Pull (CCR) SP ← (SP) + 1; Pull (A) SP ← (SP) + 1; Pull (X) SP ← (SP) + 1; Pull (PCH) SP ← (SP) + 1; Pull (PCL)	↕	↕	↕	↕	↕	INH	80		9
RTS	Return from Subroutine	SP ← (SP) + 1; Pull (PCH) SP ← (SP) + 1; Pull (PCL)	—	—	—	—	—	INH	81		6
SBC #*opr* SBC *opr* SBC *opr* SBC *opr*,X SBC *opr*,X SBC ,X	Subtract Memory Byte and Carry Bit from Accumulator	A ← (A) – (M) – (C)	—	—	↕	↕	↕	IMM DIR EXT IX2 IX1 IX	A2 B2 C2 D2 E2 F2	ii dd hh ll ee ff ff 	2 3 4 5 4 3
SEC	Set Carry Bit	C ← 1	—	—	—	—	1	INH	99		2
SEI	Set Interrupt Mask	I ← 1	—	1	—	—	—	INH	9B		2
STA *opr* STA *opr* STA *opr*,X STA *opr*,X STA ,X	Store Accumulator in Memory	M ← (A)	—	—	↕	↕	—	DIR EXT IX2 IX1 IX	B7 C7 D7 E7 F7	dd hh ll ee ff ff 	4 5 6 5 4
STOP	Stop Oscillator and Enable IRQ Pin		—	0	—	—	—	INH	8E		2
STX *opr* STX *opr* STX *opr*,X STX *opr*,X STX ,X	Store Index Register In Memory	M ← (X)	—	—	↕	↕	—	DIR EXT IX2 IX1 IX	BF CF DF EF FF	dd hh ll ee ff ff 	4 5 6 5 4
SUB #*opr* SUB *opr* SUB *opr* SUB *opr*,X SUB *opr*,X SUB ,X	Subtract Memory Byte from Accumulator	A ← (A) – (M)	—	—	↕	↕	↕	IMM DIR EXT IX2 IX1 IX	A0 B0 C0 D0 E0 F0	ii dd hh ll ee ff ff 	2 3 4 5 4 3
SWI	Software Interrupt	PC ← (PC) + 1; Push (PCL) SP ← (SP) – 1; Push (PCH) SP ← (SP) – 1; Push (X) SP ← (SP) – 1; Push (A) SP ← (SP) – 1; Push (CCR) SP ← (SP) – 1; I ← 1 PCH ← Interrupt Vector High Byte PCL ← Interrupt Vector Low Byte	—	1	—	—	—	INH	83		1 0
TAX	Transfer Accumulator to Index Register	X ← (A)	—	—	—	—	—	INH	97		2

Table 12-6. Instruction Set Summary (Sheet 6 of 6)

Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
TST *opr* TSTA TSTX TST *opr*,X TST ,X	Test Memory Byte for Negative or Zero	(M) – $00	—	—	↕	↕	—	DIR INH INH IX1 IX	3D 4D 5D 6D 7D	dd ff 	4 3 3 5 4
TXA	Transfer Index Register to Accumulator	A ← (X)	—	—	—	—	—	INH	9F		2
WAIT	Stop CPU Clock and Enable Interrupts		—	↕	—	—	—	INH	8F		2

A	Accumulator	*opr*	Operand (one or two bytes)
C	Carry/borrow flag	PC	Program counter
CCR	Condition code register	PCH	Program counter high byte
dd	Direct address of operand	PCL	Program counter low byte
dd rr	Direct address of operand and relative offset of branch instruction	REL	Relative addressing mode
DIR	Direct addressing mode	*rel*	Relative program counter offset byte
ee ff	High and low bytes of offset in indexed, 16-bit offset addressing	rr	Relative program counter offset byte
EXT	Extended addressing mode	SP	Stack pointer
ff	Offset byte in indexed, 8-bit offset addressing	X	Index register
H	Half-carry flag	Z	Zero flag
hh ll	High and low bytes of operand address in extended addressing	#	Immediate value
I	Interrupt mask	∧	Logical AND
ii	Immediate operand byte	∨	Logical OR
IMM	Immediate addressing mode	⊕	Logical EXCLUSIVE OR
INH	Inherent addressing mode	( )	Contents of
IX	Indexed, no offset addressing mode	–( )	Negation (two's complement)
IX1	Indexed, 8-bit offset addressing mode	←	Loaded with
IX2	Indexed, 16-bit offset addressing mode	?	If
M	Memory location	:	Concatenated with
N	Negative flag	↕	Set or cleared
n	Any bit	—	Not affected

12.6 Opcode Map

See **Table 12-7**.

Table 12-7. Opcode Map

	Bit Manipulation		Branch	Read-Modify-Write					Control		Register/Memory						
	DIR	DIR	REL	DIR	INH	INH	IX1	IX	INH	INH	IMM	DIR	EXT	IX2	IX1	IX	
MSB / LSB	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	MSB / LSB
0	5 BRSET0 3 DIR	5 BSET0 2 DIR	3 BRA 2 REL	5 NEG 2 DIR	3 NEGA 1 INH	3 NEGX 1 INH	6 NEG 2 IX1	5 NEG 1 IX	9 RTI 1 INH		2 SUB 2 IMM	3 SUB 2 DIR	4 SUB 3 EXT	5 SUB 3 IX2	4 SUB 2 IX1	3 SUB 1 IX	0
1	5 BRCLR0 3 DIR	5 BCLR0 2 DIR	3 BRN 2 REL						6 RTS 1 INH		2 CMP 2 IMM	3 CMP 2 DIR	4 CMP 3 EXT	5 CMP 3 IX2	4 CMP 2 IX1	3 CMP 1 IX	1
2	5 BRSET1 3 DIR	5 BSET1 2 DIR	3 BHI 2 REL		11 MUL 1 INH						2 SBC 2 IMM	3 SBC 2 DIR	4 SBC 3 EXT	5 SBC 3 IX2	4 SBC 2 IX1	3 SBC 1 IX	2
3	5 BRCLR1 3 DIR	5 BCLR1 2 DIR	3 BLS 2 REL	5 COM 2 DIR	3 COMA 1 INH	3 COMX 1 INH	6 COM 2 IX1	5 COM 1 IX	10 SWI 1 INH		2 CPX 2 IMM	3 CPX 2 DIR	4 CPX 3 EXT	5 CPX 3 IX2	4 CPX 2 IX1	3 CPX 1 IX	3
4	5 BRSET2 3 DIR	5 BSET2 2 DIR	3 BCC 2 REL	5 LSR 2 DIR	3 LSRA 1 INH	3 LSRX 1 INH	6 LSR 2 IX1	5 LSR 1 IX			2 AND 2 IMM	3 AND 2 DIR	4 AND 3 EXT	5 AND 3 IX2	4 AND 2 IX1	3 AND 1 IX	4
5	5 BRCLR2 3 DIR	5 BCLR2 2 DIR	3 BCS/BLO 2 REL								2 BIT 2 IMM	3 BIT 2 DIR	4 BIT 3 EXT	5 BIT 3 IX2	4 BIT 2 IX1	3 BIT 1 IX	5
6	5 BRSET3 3 DIR	5 BSET3 2 DIR	3 BNE 2 REL	5 ROR 2 DIR	3 RORA 1 INH	3 RORX 1 INH	6 ROR 2 IX1	5 ROR 1 IX			2 LDA 2 IMM	3 LDA 2 DIR	4 LDA 3 EXT	5 LDA 3 IX2	4 LDA 2 IX1	3 LDA 1 IX	6
7	5 BRCLR3 3 DIR	5 BCLR3 2 DIR	3 BEQ 2 REL	5 ASR 2 DIR	3 ASRA 1 INH	3 ASRX 1 INH	6 ASR 2 IX1	5 ASR 1 IX		2 TAX 1 INH		4 STA 2 DIR	5 STA 3 EXT	6 STA 3 IX2	5 STA 2 IX1	4 STA 1 IX	7
8	5 BRSET4 3 DIR	5 BSET4 2 DIR	3 BHCC 2 REL	5 ASL/LSL 2 DIR	3 ASLA/LSLA 1 INH	3 ASLX/LSLX 1 INH	6 ASL/LSL 2 IX1	5 ASL/LSL 1 IX		2 CLC 1 INH	2 EOR 2 IMM	3 EOR 2 DIR	4 EOR 3 EXT	5 EOR 3 IX2	4 EOR 2 IX1	3 EOR 1 IX	8
9	5 BRCLR4 3 DIR	5 BCLR4 2 DIR	3 BHCS 2 REL	5 ROL 2 DIR	3 ROLA 1 INH	3 ROLX 1 INH	6 ROL 2 IX1	5 ROL 1 IX		2 SEC 1 INH	2 ADC 2 IMM	3 ADC 2 DIR	4 ADC 3 EXT	5 ADC 3 IX2	4 ADC 2 IX1	3 ADC 1 IX	9
A	5 BRSET5 3 DIR	5 BSET5 2 DIR	3 BPL 2 REL	5 DEC 2 DIR	3 DECA 1 INH	3 DECX 1 INH	6 DEC 2 IX1	5 DEC 1 IX		2 CLI 1 INH	2 ORA 2 IMM	3 ORA 2 DIR	4 ORA 3 EXT	5 ORA 3 IX2	4 ORA 2 IX1	3 ORA 1 IX	A
B	5 BRCLR5 3 DIR	5 BCLR5 2 DIR	3 BMI 2 REL							2 SEI 1 INH	2 ADD 2 IMM	3 ADD 2 DIR	4 ADD 3 EXT	5 ADD 3 IX2	4 ADD 2 IX1	3 ADD 1 IX	B
C	5 BRSET6 3 DIR	5 BSET6 2 DIR	3 BMC 2 REL	5 INC 2 DIR	3 INCA 1 INH	3 INCX 1 INH	6 INC 2 IX1	5 INC 1 IX		2 RSP 1 INH		2 JMP 2 DIR	3 JMP 3 EXT	4 JMP 3 IX2	3 JMP 2 IX1	2 JMP 1 IX	C
D	5 BRCLR6 3 DIR	5 BCLR6 2 DIR	3 BMS 2 REL	4 TST 2 DIR	3 TSTA 1 INH	3 TSTX 1 INH	5 TST 2 IX1	4 TST 1 IX		2 NOP 1 INH	6 BSR 2 REL	5 JSR 2 DIR	6 JSR 3 EXT	7 JSR 3 IX2	6 JSR 2 IX1	5 JSR 1 IX	D
E	5 BRSET7 3 DIR	5 BSET7 2 DIR	3 BIL 2 REL						2 STOP 1 INH		2 LDX 2 IMM	3 LDX 2 DIR	4 LDX 3 EXT	5 LDX 3 IX2	4 LDX 2 IX1	3 LDX 1 IX	E
F	5 BRCLR7 3 DIR	5 BCLR7 2 DIR	3 BIH 2 REL	5 CLR 2 DIR	3 CLRA 1 INH	3 CLRX 1 INH	6 CLR 2 IX1	5 CLR 1 IX	2 WAIT 1 INH	2 TXA 1 INH		4 STX 2 DIR	5 STX 3 EXT	6 STX 3 IX2	5 STX 2 IX1	4 STX 1 IX	F

INH = Inherent
IMM = Immediate
DIR = Direct
EXT = Extended

REL = Relative
IX = Indexed, No Offset
IX1 = Indexed, 8-Bit Offset
IX2 = Indexed, 16-Bit Offset

LSB of Opcode in Hexadecimal

MSB / LSB	0
0	5 BRSET0 3 DIR

MSB of Opcode in Hexadecimal
Number of Cycles
Opcode Mnemonic
Number of Bytes/Addressing Mode

Section 13. Electrical Specifications

13.1 Contents

13.2 Introduction

This section contains electrical and timing specifications.

13.3 Maximum Ratings

Maximum ratings are the extreme limits to which the MCU can be exposed without permanently damaging it.

The MCU contains circuitry to protect the inputs against damage from high static voltages; however, do not apply voltages higher than those shown in the table here. Keep V_{In} and V_{Out} within the range $V_{SS} \le (V_{In}$ or $V_{Out}) \le V_{DD}$. Connect unused inputs to the appropriate voltage level, either V_{SS} or V_{DD}.

Rating[1]	Symbol	Value	Unit
Supply voltage	V_{DD}	–0.3 to +7.0	V
Input voltage	V_{In}	V_{SS} –0.3 to V_{DD} +0.3	V
Programming voltage	V_{PP}	V_{DD} –0.3 to 16.0	
Bootstrap mode ($\overline{IRQ}$ pin only)	V_{In}	V_{SS} – 0.3 to 2 x V_{DD} + 0.3	V
Current drain per pin excluding V_{DD} and V_{SS}	I	25	mA
Storage temperature range	T_{STG}	–65 to +150	°C

1. Voltages referenced to V_{SS}

NOTE: *This device is not guaranteed to operate properly at the maximum ratings. Refer to* ***13.7 5.0-Volt DC Electrical Characteristics*** *and* ***13.8 3.3-Volt DC Electrical Characteristics*** *for guaranteed operating conditions.*

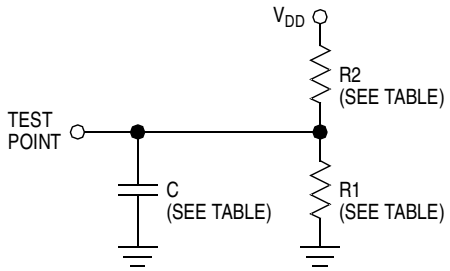
13.4 Operating Temperature Range

Rating(1)	Symbol	Value	Unit
Operating temperature range(2) MC68HC705C8ACB MC68HC705C8ACFB MC68HC705C8ACFS MC68HC705C8ACP MC68HC705C8ACFN MC68HC705C8ACFS	T_A	T_L to T_H – 40 to + 85	°C

1. Voltages referenced to V_{SS}
2. C = Extended temperature range (– 40°C to + 85°C)
   P = Plastic dual in-line package (PDIP)
   B = Plastic shrink dual in-line package (SDIP)
   S = Ceramic dual in-line package (cerdip)
   FN = Plastic-leaded chip carrier (PLCC)
   FB = Quad flat pack (QFP)
   FS = Ceramic-leaded chip carrier (CLCC)

13.5 Thermal Characteristics

Characteristic	Symbol	Value	Unit
Thermal resistance Plastic dual in-line package (DIP) Ceramic dual in-line package (cerdip) Plastic leaded chip carrier (PLCC) Quad flat pack (QFP) Plastic shrink DIP (SDIP)	θ_{JA}	 60 50 70 95 60	°C/W



V_{DD} = 4.5 V

Pins	R1	R2	C
PA7–PA0 PB7–PB0 PC7–PC0 PD4–PD1	3.26 kΩ	2.38 kΩ	50 pF

V_{DD} = 3.0 V

Pins	R1	R2	C
PA7–PA0 PB7–PB0 PC7–PC0 PD4–PD1	10.91 kΩ	6.32 kΩ	50 pF
PD7, PD5, PD0	6 kΩ	6 kΩ	200 pF

Figure 13-1. Equivalent Test Load

13.6 Power Considerations

The average chip junction temperature, T_J, in °C can be obtained from:

$$T_J = T_A + (P_D \times \theta_{JA}) \quad (1)$$

Where:

T_A = ambient temperature in °C
θ_{JA} = package thermal resistance, junction to ambient in °C/W
$P_D = P_{INT} + P_{I/O}$
$P_{INT} = I_{CC} \times V_{CC}$ = chip internal power dissipation
$P_{I/O}$ = power dissipation on input and output pins (user-determined)

For most applications, $P_{I/O} < P_{INT}$ and can be neglected.

Ignoring $P_{I/O}$, the relationship between P_D and T_J is approximately:

$$P_D = \frac{K}{T_J + 273°C} \quad (2)$$

Solving equations (1) and (2) for K gives:

$$= P_D \times (T_A + 273°C) + \theta_{JA} \times (P_D)^2 \quad (3)$$

where K is a constant pertaining to the particular part. K can be determined from equation (3) by measuring P_D (at equilibrium) for a known T_A. Using this value of K, the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

13.7 5.0-Volt DC Electrical Characteristics

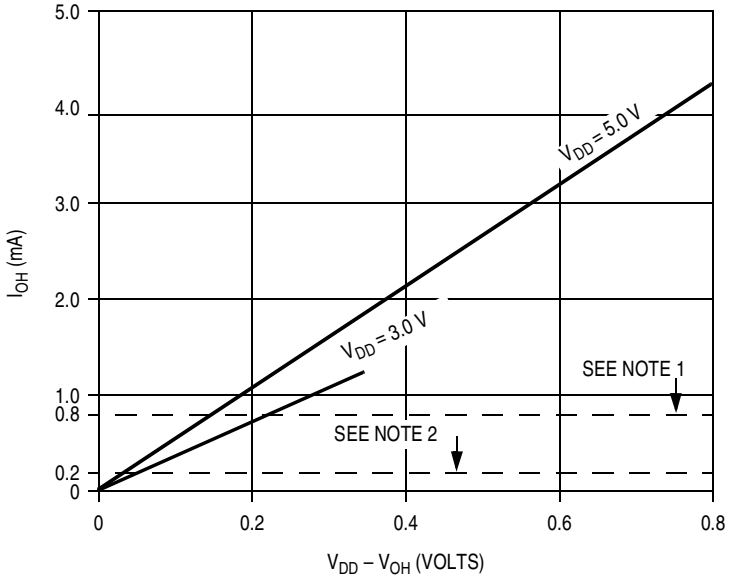
Characteristic[1]	Symbol	Min	Typ[2]	Max	Unit
Output voltage, $I_{Load} \leq 10.0\ \mu A$	V_{OL} V_{OH}	— $V_{DD} - 0.1$	— —	0.1 —	V
Output high voltage I_{Load} = –0.8 mA, PA7–PA0, PB7–PB0, PC6–PC0, TCMP (see **Figure 13-2**) I_{Load} = –1.6 mA, PD4–PD1 (see **Figure 13-3**) I_{Load} = –5.0 mA, PC7	V_{OH}	$V_{DD} - 0.8$	— — —	— — —	V
Output low voltage (see **Figure 13-4**) I_{Load} = 1.6 mA PA7–PA0, PB7–PB0, PC6–PC0, PD4–PD1 I_{Load} = 20 mA, PC7	V_{OL}	— —	— —	0.4 0.4	V
Input high voltage PA7–PA0, PB7–PB0, PC7–PC0, PD5–PD0, PD7, TCAP, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IH}	$0.7 \times V_{DD}$	—	V_{DD}	V
Input low voltage PA7–PA0, PB7–PB0, PC7–PC0, PD5–PD0, PD7, TCAP, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IL}	V_{SS}	—	$0.2 \times V_{DD}$	V
EPROM programming voltage	V_{PP}	14.5	14.75	15.0	V
EPROM/OTPROM programming current	I_{PP}	—	5	10	mA
User mode current	I_{PP}	—	—	± 10	mA
Data-retention mode (0°C to 70°C)	V_{RM}	2.0	—	—	V
Supply current [3] Run[4] Wait[5] Stop[6] 25°C –40°C to +85°C	I_{DD}	 — — — —	 5.0 1.95 5.0 5.0	 7.0 3.0 50 50	 mA mA µµA µµA
I/O ports hi-z leakage current PA7–PA0, PB7–PB0, PC7–PC0, PD4–PD1, PD7, $\overline{RESET}$	I_{IL}	—	—	± 10	µA
Input current, $\overline{IRQ}$, TCAP, OSC1, PD0, PD5	I_{In}	—	—	± 1	µA
Capacitance Ports (as input or output) $\overline{RESET}$, $\overline{IRQ}$, TCAP, PD0–PD5, PD7	C_{Out} C_{In}	— —	— —	12 8	pF

1. V_{DD} = 5 V ± 10%; V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, unless otherwise noted
2. Typical values reflect average measurements at midpoint of voltage range at 25°C.
3. I_{DD} measured with port B pullup devices disabled.
4. Run (operating) I_{DD} measured using external square wave clock source (f_{OSC} = 4.2 MHz). All inputs 0.2 V from rail. No dc loads. Less than 50 pF on all outputs. C_L = 20 pF on OSC2. OSC2 capacitance linearly affects run I_{DD}.
5. Wait I_{DD} measured using external square wave clock source (f_{OSC} = 4.2 MHz). All inputs 0.2 V from rail. No dc loads. Less than 50 pF on all outputs. C_L = 20 pF on OSC2. V_{IL} = 0.2 V, $V_{IH} = V_{DD} - 0.2$ V. All ports configured as inputs. SPI and SCI disabled. If SPI and SCI enabled, add 10% current draw. OSC2 capacitance linearly affects wait I_{DD}.
6. Stop I_{DD} measured with OSC1 = V_{DD}. All ports configured as inputs. V_{IL} = 0.2 V, $V_{IH} = V_{DD} - 0.2$ V.

13.8 3.3-Volt DC Electrical Characteristics

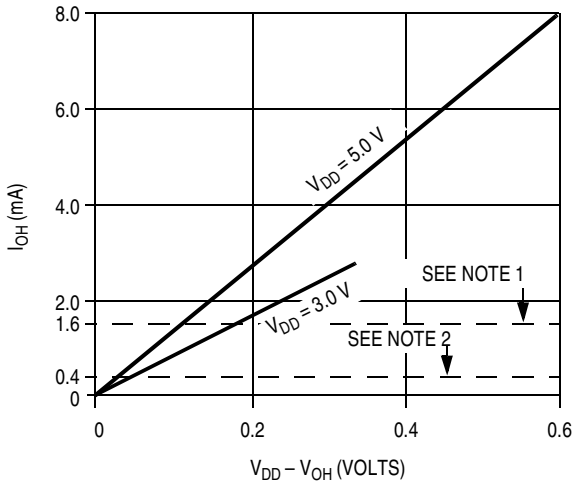
Characteristic[1]	Symbol	Min	Typ[2]	Max	Unit
Output voltage, $I_{Load} \leq 10.0\ \mu A$	V_{OL} V_{OH}	— $V_{DD} - 0.1$	— —	0.1 —	V
Output high voltage $I_{Load} = -0.2$ mA PA7–PA0, PB7–PB0, PC6–PC0, TCMP (see **Figure 13-2**) $I_{Load} = -0.4$ mA PD4–PD1 (see **Figure 13-3**) $I_{Load} = -1.5$ mA PC7	V_{OH}	$V_{DD} - 0.3$	— — —	— — —	V
Output low voltage (see **Figure 13-4**) $I_{Load} = 0.4$ mA PA7–PA0, PB7–PB0, PC6–PC0, PD4–PD1 $I_{Load} = 6.0$ mA PC7	V_{OL}	— —	— —	0.3 0.3	V
Input high voltage PA7–PA0, PB7–PB0, PC7–PC0, PD5–PD0, PD7, TCAP, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IH}	$0.7 \times V_{DD}$	—	V_{DD}	V
Input low voltage PA7–PA0, PB7–PB0, PC7–PC0, PD5–PD0, PD7, TCAP, $\overline{IRQ}$, $\overline{RESET}$, OSCI	V_{IL}	V_{SS}	—	$0.2 \times V_{DD}$	V
Data-retention mode (0°C to 70°C)	V_{RM}	2.0	—	—	V
Supply current[3] Run[4] Wait[5] Stop[6]	I_{DD}	— — —	1.53 0.711 2.0	3.0 1.0 20	mA mA µA
I/O ports hi-z leakage current PA7–PA0, PB7–PB0, PC7–PC0, PD4–PD1, PD7, $\overline{RESET}$	I_{IL}	—	—	± 10	µA
Input current $\overline{IRQ}$, TCAP, OSC1, PD5, PD0	I_{In}	—	—	± 1	µA

1. $V_{DD} = 3.3\ V \pm 10\%$; $V_{SS} = 0$ Vdc, $T_A = T_L$ to T_H, unless otherwise noted
2. Typical values at midpoint of voltage range, 25°C only.
3. I_{DD} measured with port B pullup devices disabled.
4. Run (operating) I_{DD} measured using external square wave clock source ($f_{OSC} = 2.0$ MHz). All inputs 0.2 V from rail. No dc loads. Less than 50 pF on all outputs. $C_L = 20$ pF on OSC2. OSC2 capacitance linearly affects run I_{DD}.
5. Wait I_{DD} measured using external square wave clock source ($f_{OSC} = 2.0$ MHz). All inputs 0.2 V from rail. No dc loads. Less than 50 pF on all outputs. $C_L = 20$ pF on OSC2. $V_{IL} = 0.2$ V, $V_{IH} = V_{DD} - 0.2$ V. All ports configured as inputs. SPI and SCI disabled. If SPI and SCI enabled, add 10% current draw. OSC2 capacitance linearly affects wait I_{DD}.
6. Stop I_{DD} measured with OSC1 = V_{DD}. All ports configured as inputs. $V_{IL} = 0.2$ V; $V_{IH} = V_{DD} - 0.2$ V.



Notes:
1. At V_{DD} = 5.0 V, devices are specified and tested for ($V_{DD} - V_{OH}$) ≤ 800 mV @ I_{OH} = –0.8 mA.
2. At V_{DD} = 3.3 V, devices are specified and tested for ($V_{DD} - V_{OH}$) ≤ 300 mV @ I_{OH} = –0.2 mA.

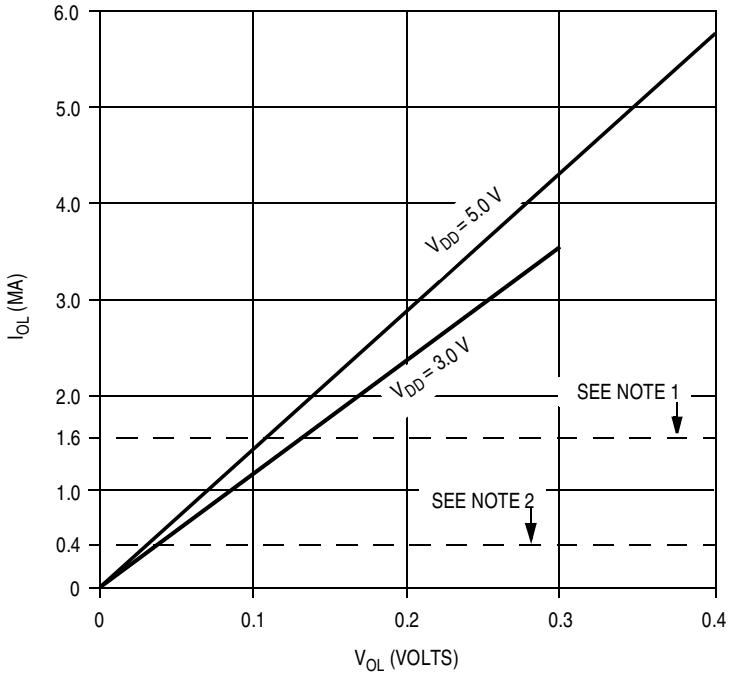
(a) V_{OH} versus I_{OH} for Ports A, B, PC6–PC0, and TCMP



Notes:
1. At V_{DD} = 5.0 V, devices are specified and tested for ($V_{DD} - V_{OH}$) ≤ 800 mV @ I_{OH} = –1.6 mA.
2. At V_{DD} = 3.3 V, devices are specified and tested for ($V_{DD} - V_{OH}$) ≤ 300 mV @ I_{OH} = –0.4 mA.

(b) V_{OH} versus I_{OH} for PD4–PD1

Figure 13-2. Typical Voltage Compared to Current

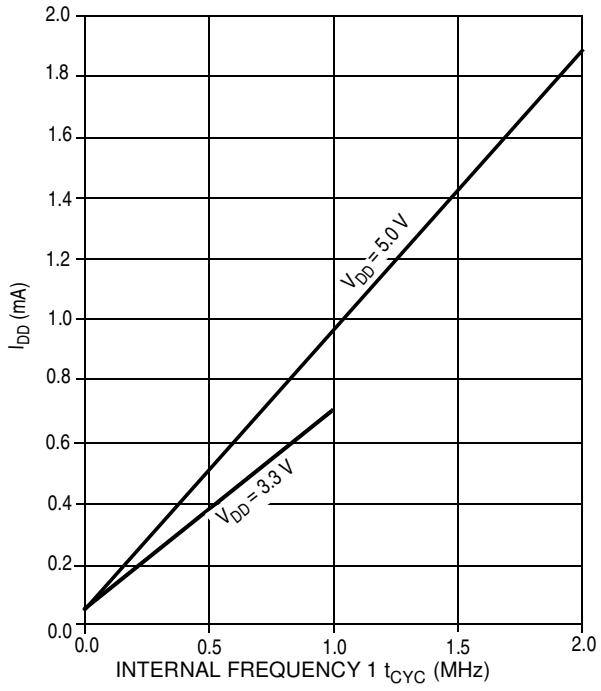


Notes:

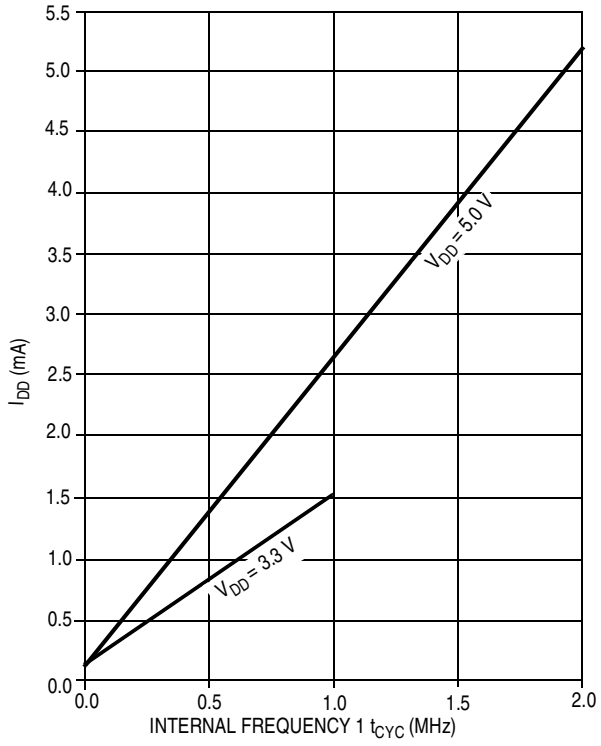
1. At V_{DD} = 5.0 V, devices are specified and tested for $V_{OL} \leq 400$ mV @ I_{OL} = 1.6 mA.
2. At V_{DD} = 3.3 V, devices are specified and tested for $V_{OL} \leq 300$ mV @ I_{OL} = 0.4 mA.

(c) V_{OL} versus I_{OL} for All Ports Except PC7

Figure 13-2. Typical Voltage Compared to Current (Continued)

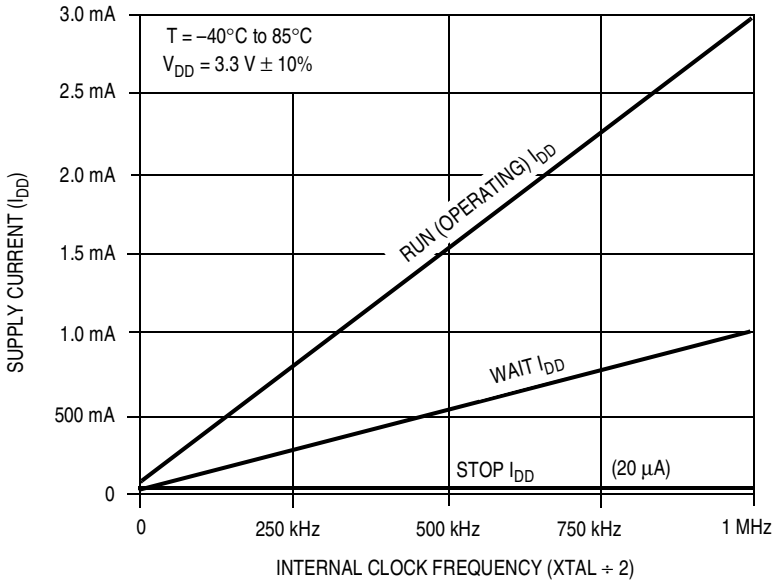


(a) Wait Mode

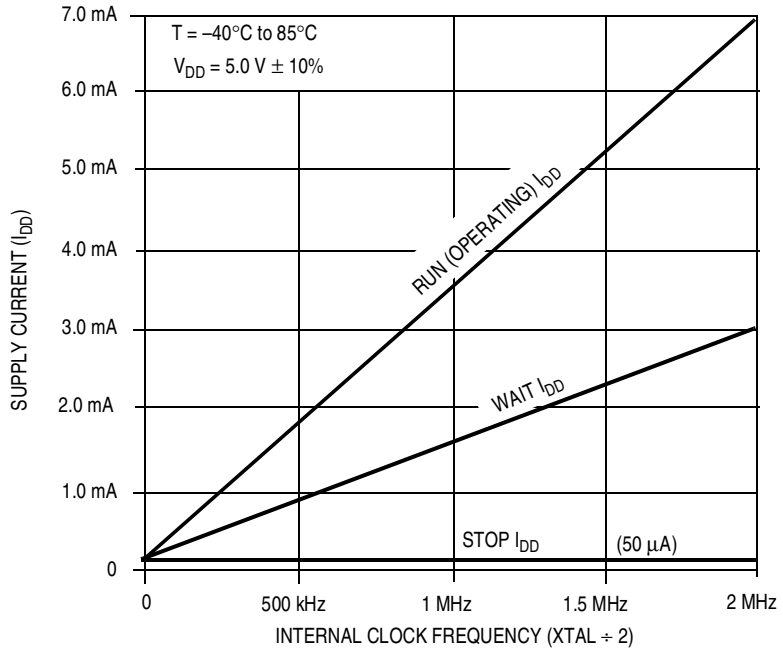


(b) Run Mode

Figure 13-3. Typical Current versus Internal Frequency for Run and Wait Modes



(a) Maximum Current Drain versus Frequency @ 3.3 V ± 10 %



(b) Maximum Current Drain versus Frequency @ 5 V ± 10%

Figure 13-4. Total Current Drain versus Frequency

13.9 5.0-Volt Control Timing

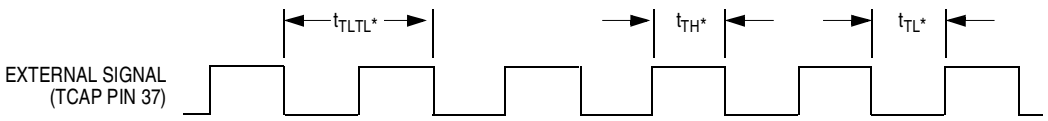
Characteristic[1]	Symbol	Min	Max	Unit
Frequency of operation Crystal option External clock option	f_{OSC}	— dc	4.2 4.2	MHz
Internal operating frequency Crystal ($f_{OSC} \div 2$) External clock ($f_{OSC} \div 2$)	f_{OP}	— dc	2.1 2.1	MHz
Cycle time (see **Figure 13-7**)	t_{CYC}	480	—	ns
Crystal oscillator startup time (see **Figure 13-7**)	t_{OXOV}	—	100	ms
Stop recovery startup time (crystal oscillator) (see **Figure 13-6**)	t_{ILCH}	—	100	ms
$\overline{\text{RESET}}$ pulse width (see **Figure 13-7**)	t_{RL}	8	—	t_{CYC}
Timer Resolution[2] Input capture pulse width (see **Figure 13-5**) Input capture pulse period (see **Figure 13-5**)	t_{RESL} t_{TH}, t_{TL} t_{TLTL}	4.0 125 [3]	— — —	t_{CYC} ns t_{CYC}
Interrupt pulse width low (edge-triggered) (see **Figure 4-2. External Interrupt Timing**)	t_{ILIH}	125	—	ns
Interrupt pulse period (see **Figure 4-2. External Interrupt Timing**)	t_{ILIL}	[4]	—	t_{CYC}
OSC1 pulse width	t_{OH}, t_{OL}	90	—	ns

1. V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc; $T_A = T_L$ to T_H
2. Since a 2-bit prescaler in the timer must count four internal cycles (t_{CYC}), this is the limiting minimum factor in determining the timer resolution.
3. The minimum period, t_{TLTL}, should not be less than the number of cycle times it takes to execute the capture interrupt service routine plus 24 t_{CYC}.
4. The minimum period, t_{ILIL}, should not be less than the number of cycle times it takes to execute the interrupt service routine plus 19 t_{CYC}.

13.10 3.3-Volt Control Timing

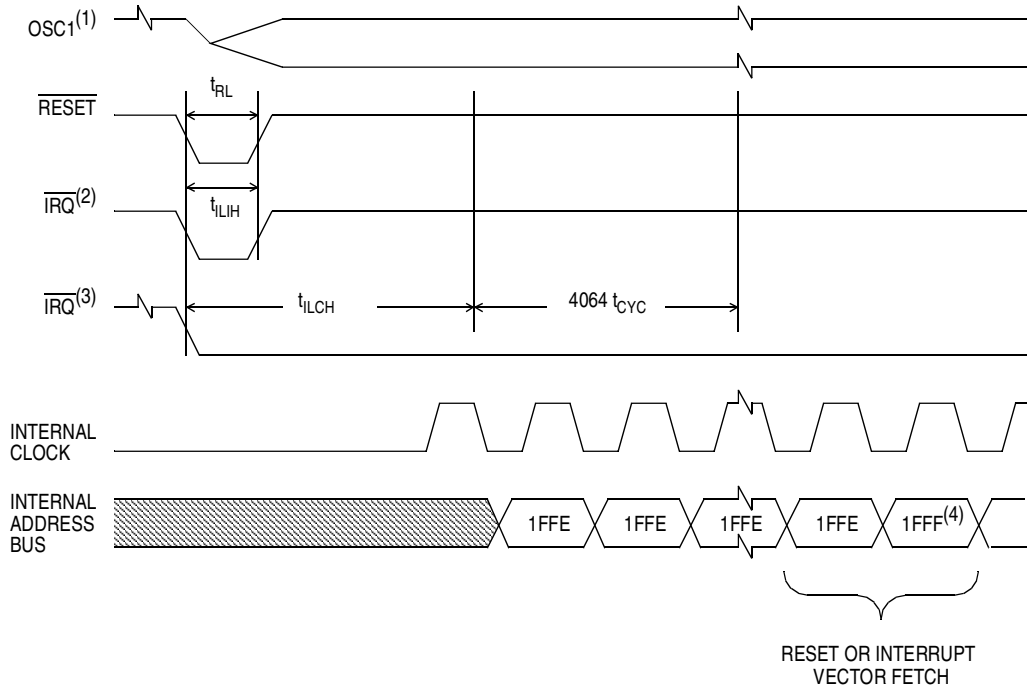
Characteristic[1]	Symbol	Min	Max	Unit
Frequency of operation Crystal option External clock option	f_{OSC}	— dc	2.0 2.0	MHz
Internal operating frequency Crystal ($f_{OSC} \div 2$) External clock ($f_{OSC} \div 2$)	f_{OP}	— dc	1.0 1.0	MHz
Cycle time (see **Figure 13-7**)	t_{CYC}	1000	—	ns
Crystal oscillator startup time (see **Figure 13-7**)	t_{OXOV}	—	100	ms
Stop recovery startup time (crystal oscillator) (see **Figure 13-6**)	t_{ILCH}	—	100	ms
$\overline{\text{RESET}}$ pulse width, excluding power-up (see **Figure 13-7**)	t_{RL}	8	—	t_{CYC}
Timer Resolution[2] Input capture pulse width (see **Figure 13-5**) Input capture pulse period (see **Figure 13-5**)	 t_{RESL} t_{TH}, t_{TL} t_{TLTL}	 4.0 250 [3]	 — — —	 t_{CYC} ns t_{CYC}
Interrupt pulse width low (edge-triggered) (see **Figure 4-2. External Interrupt Timing**)	t_{ILIH}	250	—	ns
Interrupt pulse period (see **Figure 4-2. External Interrupt Timing**)	t_{ILIL}	[4]	—	t_{CYC}
OSC1 pulse width	t_{OH}, t_{OL}	200	—	ns

1. V_{DD} = 3.3 Vdc ± 0.3 Vdc, V_{SS} = 0 Vdc; $T_A = T_L$ to T_H
2. Since a 2-bit prescaler in the timer must count four internal cycles (t_{CYC}), this is the limiting minimum factor in determining the timer resolution.
3. The minimum period, t_{TLTL}, should not be less than the number of cycle times it takes to execute the capture interrupt service routine plus 24 t_{CYC}.
4. The minimum period, t_{ILIL}, should not be less than the number of cycle times it takes to execute the interrupt service routine plus 19 t_{CYC}.



*Refer to timer resolution data in **Figure 13-6** and **Figure 13-7**.

Figure 13-5. Timer Relationships



Notes:
1. Represents the internal gating of the OSC1 pin
2. $\overline{\text{IRQ}}$ pin edge-sensitive option
3. $\overline{\text{IRQ}}$ pin level and edge-sensitive option
4. $\overline{\text{RESET}}$ vector address shown for timing example

Figure 13-6. Stop Recovery Timing Diagram

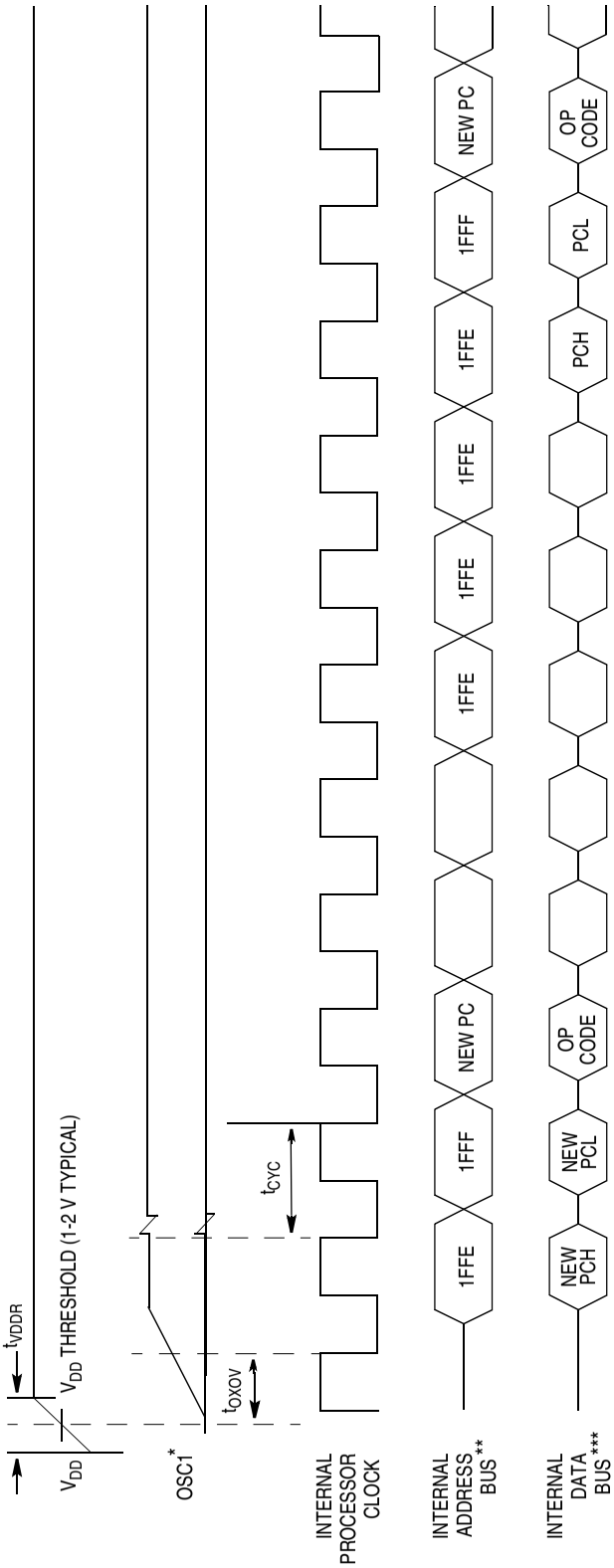


Figure 13-7. Power-On Reset and External Reset Timing Diagram

* OSC1 line is not meant to represent frequency. It is only used to represent time.

** Internal timing signal and bus information are not available externally.

*** The next rising edge of the internal processor clock following the rising edge of $\overline{\text{RESET}}$ initiates the reset sequence.

13.11 5.0-Volt Serial Peripheral Interface (SPI) Timing

Number[1]	Characteristic[2]	Symbol	Min	Max	Unit
	Operating frequency Master Slave	$f_{OP(M)}$ $f_{OP(S)}$	dc dc	0.5 2.1	f_{OP} MHz
1	Cycle time Master Slave	$t_{CYC(M)}$ $t_{CYC(S)}$	2.0 480	— —	t_{CYC} ns
2	Enable lead time Master Slave	$t_{Lead(M)}$ $t_{Lead(S)}$	(3) 240	— —	ns
3	Enable lag time Master Slave	$t_{Lag(M)}$ $t_{Lag(S)}$	(2) 720	— —	ns
4	Clock (SCK) high time Master Slave	$t_{W(SCKH)M}$ $t_{W(SCKH)S}$	340 190	— —	ns
5	Clock (SCK) low time Master Slave	$t_{W(SCKL)M}$ $t_{W(SCKL)S}$	340 190	— —	ns
6	Data setup time (inputs) Master Slave	$t_{SU(M)}$ $t_{SU(S)}$	100 100	— —	ns
7	Data hold time (inputs) Master Slave	$t_{H(M)}$ $t_{H(S)}$	100 100	— —	ns
8	Access time[4] Slave	t_A	0	120	ns
9	Disable time[5] Slave	t_{DIS}	—	240	ns
10	Data valid time Master (before capture edge) Slave (after enable edge)[6]	$t_{V(M)}$ $t_{V(S)}$	0.25 —	— 240	$t_{CYC(M)}$ ns

Continued

Number[1]	Characteristic[2]	Symbol	Min	Max	Unit
11	Data hold time (outputs) Master (after capture edge) Slave (after enable edge)	 $t_{HO(M)}$ $t_{HO(S)}$	 0.25 0	 — —	 $t_{CYC(M)}$ ns
12	Rise time[7] SPI outputs (SCK, MOSI, MISO) SPI inputs (SCK, MOSI, MISO, $\overline{SS}$)	 $t_{R(M)}$ $t_{R(S)}$	 — —	 100 2.0	 ns µs
13	Fall time[8] SPI outputs (SCK, MOSI, MISO) SPI inputs (SCK, MOSI, MISO, $\overline{SS}$)	 $t_{F(M)}$ $t_{F(S)}$	 — —	 100 2.0	 ns µs

1. Numbers refer to dimensions in **Figure 13-8** and **Figure 13-9**.
2. V_{DD} = 5.0 Vdc ± 10%
3. Signal production depends on software.
4. Time to data active from high-impedance state
5. Hold time to high-impedance state
6. With 200 pF on all SPI pins
7. 20% of V_{DD} to 70% of V_{DD}; C_L = 200 pF
8. 70% of V_{DD} to 20% of V_{DD}; C_L = 200 pF

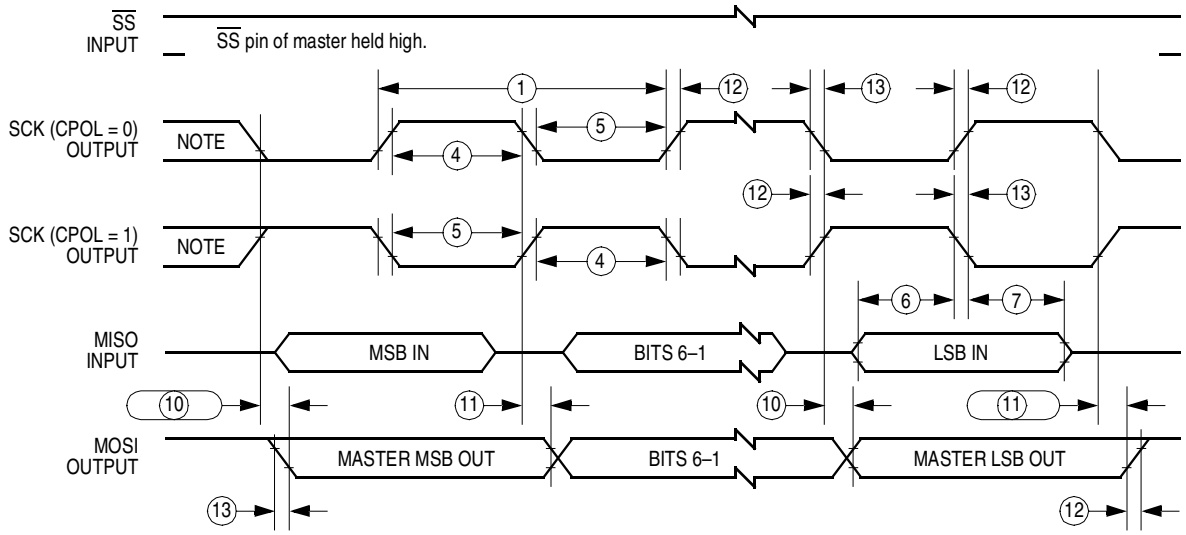
13.12 3.3-Volt Serial Peripheral Interface (SPI) Timing

Number[1]	Characteristic[2]	Symbol	Min	Max	Unit
	Operating frequency Master Slave	$f_{OP(M)}$ $f_{OP(S)}$	dc	0.5 2.1	f_{OP} MHz
1	Cycle time Master Slave	$t_{CYC(M)}$ $t_{CYC(S)}$	2.0 1	— —	t_{CYC} ns
2	Enable lead time Master Slave	$t_{Lead(M)}$ $t_{Lead(S)}$	(3) 500	— —	ns
3	Enable lag time Master Slave	$t_{Lag(M)}$ $t_{Lag(S)}$	(2) 1500	— —	ns
4	Clock (SCK) high time Master Slave	$t_{W(SCKH)M}$ $t_{W(SCKH)S}$	720 400	— —	ns
5	Clock (SCK) low time Master Slave	$t_{W(SCKL)M}$ $t_{W(SCKL)S}$	720 400	— —	ns
6	Data setup time (inputs) Master Slave	$t_{SU(M)}$ $t_{SU(S)}$	200 200	— —	ns
7	Data hold time (inputs) Master Slave	$t_{H(M)}$ $t_{H(S)}$	200 200	— —	ns
8	Access time[4] Slave	t_A	0	250	ns
9	Disable time[5] Slave	t_{DIS}	—	500	ns
10	Data valid time Master (before capture edge) Slave (after enable edge)[6]	$t_{V(M)}$ $t_{V(S)}$	0.25 —	— 500	$t_{CYC(M)}$ ns

Continued

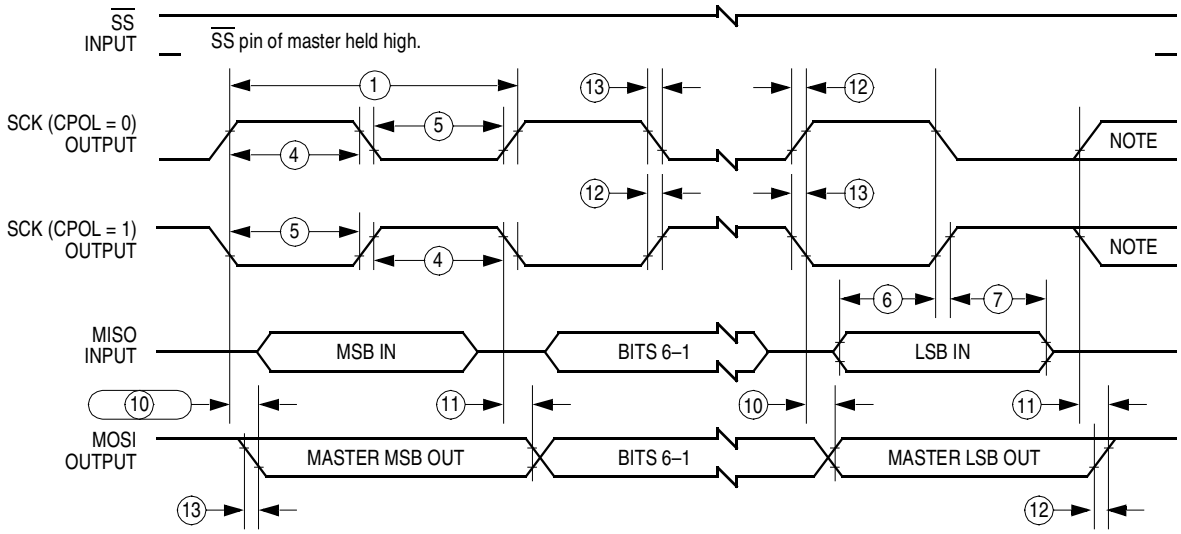
Number[1]	Characteristic[2]	Symbol	Min	Max	Unit
11	Data hold time (outputs) Master (after capture edge) Slave (after enable edge)	 $t_{HO(M)}$ $t_{HO(S)}$	 0.25 0	 — —	$t_{CYC(M)}$ ns
12	Rise time[7] SPI outputs (SCK, MOSI, MISO) SPI inputs (SCK, MOSI, MISO, $\overline{SS}$)	 $t_{R(M)}$ $t_{R(S)}$	 — —	 200 2.0	 ns µs
13	Fall time[8] SPI outputs (SCK, MOSI, MISO) SPI inputs (SCK, MOSI, MISO, $\overline{SS}$)	 $t_{F(M)}$ $t_{F(S)}$	 — —	 200 2.0	 ns µs

1. Numbers refer to dimensions in **Figure 13-8** and **Figure 13-9**.
2. V_{DD} = 3.3 Vdc ± 10%
3. Signal production depends on software.
4. Time to data active from high-impedance state
5. Hold time to high-impedance state
6. With 200 pF on all SPI pins
7. 20% of V_{DD} to 70% of V_{DD}; C_L = 200 pF
8. 70% of V_{DD} to 20% of V_{DD}; C_L = 200 pF



Note: This first clock edge is generated internally, but is not seen at the SCK pin.

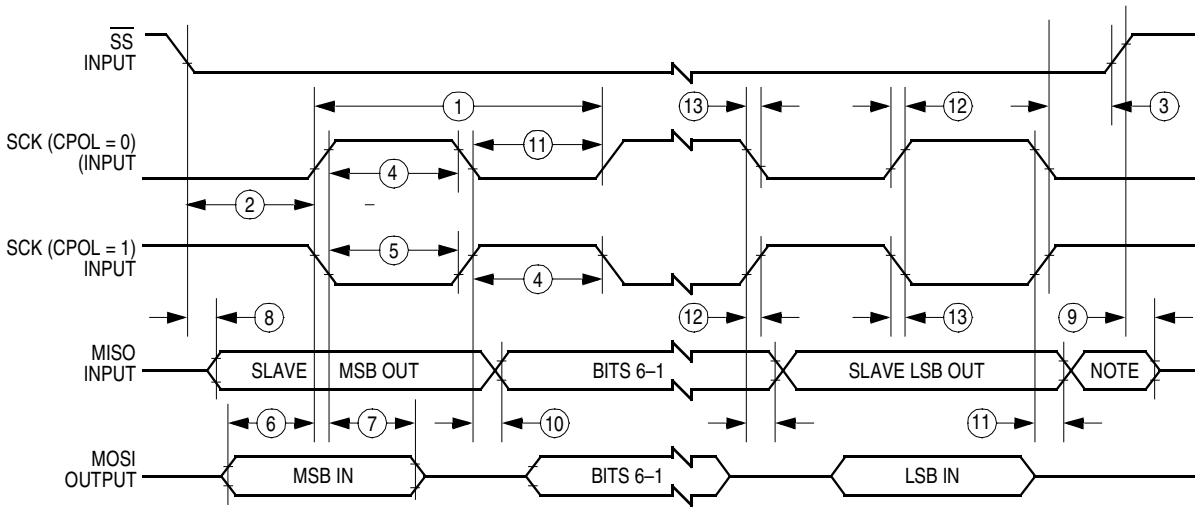
a) SPI Master Timing (CPHA = 0)



Note: This last clock edge is generated internally, but is not seen at the SCK pin.

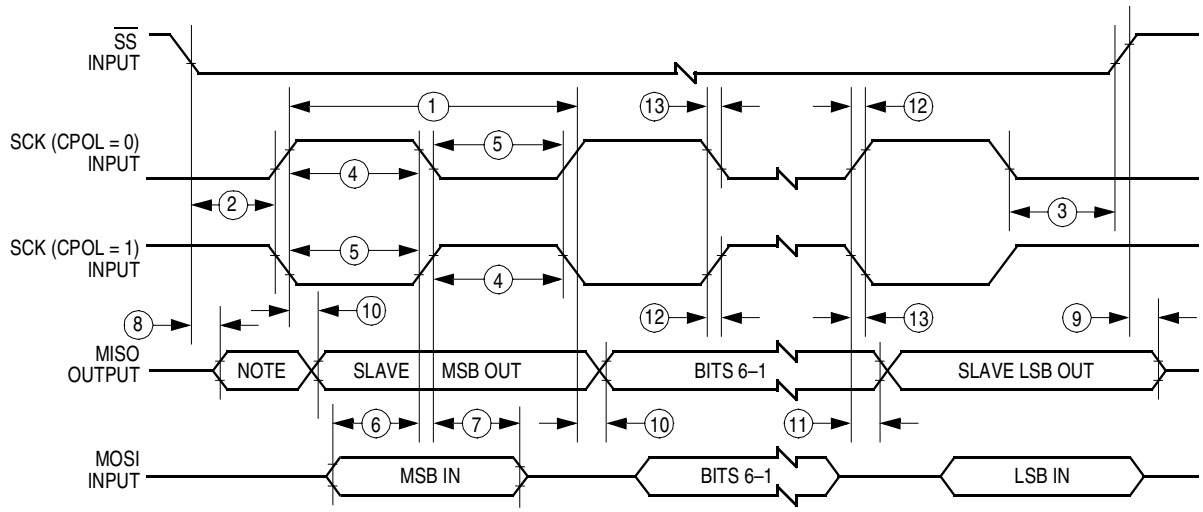
b) SPI Master Timing (CPHA = 1)

Figure 13-8. SPI Master Timing



Note: Not defined, but normally MSB of character just received

a) SPI Slave Timing (CPHA = 0)



Note: Not defined, but normally LSB of character previously transmitted

b) SPI Slave Timing (CPHA = 1)

Figure 13-9. SPI Slave Timing

Section 14. Mechanical Specifications

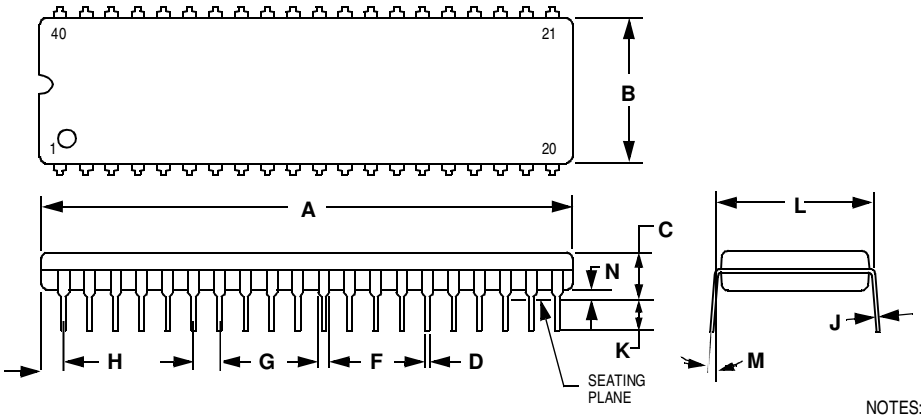
14.1 Contents

14.2 Introduction

Package dimensions available at the time of this publication for the MC68HC705C8A are provided in this section. The packages are:

- 40-pin plastic dual in-line package (PDIP)
- 40-pin ceramic dual-in-line package (cerdip)
- 44-lead plastic-leaded chip carrier (PLCC)
- 44-lead ceramic-leaded chip carrier (CLCC)
- 44-pin quad flat pack (QFP)
- 42-pin shrink dual in-line package (SDIP)

14.3 40-Pin Plastic Dual In-Line Package (PDIP)

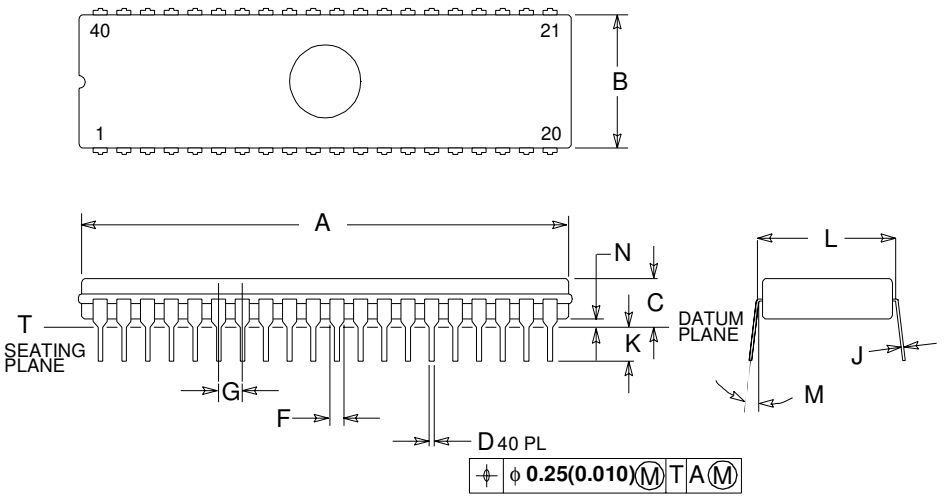


DIM	MILLIMETERS		INCHES	
	MIN	MAX	MIN	MAX
A	51.69	52.45	2.035	2.065
B	13.72	14.22	0.540	0.560
C	3.94	5.08	0.155	0.200
D	0.36	0.56	0.014	0.022
F	1.02	1.52	0.040	0.060
G	2.54 BSC		0.100 BSC	
H	1.65	2.16	0.065	0.085
J	0.20	0.38	0.008	0.015
K	2.92	3.43	0.115	0.135
L	15.24 BSC		0.600 BSC	
M	0°	1°	0°	1°
N	0.51	1.02	0.020	0.040

NOTES:
1. POSITION TOLERANCE OF LEADS (D), SHALL BEWITHIN 0.25 (0.010) AT MAXIMUM MATERIAL CONDITIONS, IN RELATION TO SEATING PLANE AND EACH OTHER.
2. DIMENSION L TO CENTER OF LEADS WHEN FORMED PARALLEL.
3. DIMENSION B DOES NOT INCLUDE MOLD FLASH.

Figure 14-1. MC68HC705C8AP Package Dimensions (Case #711)

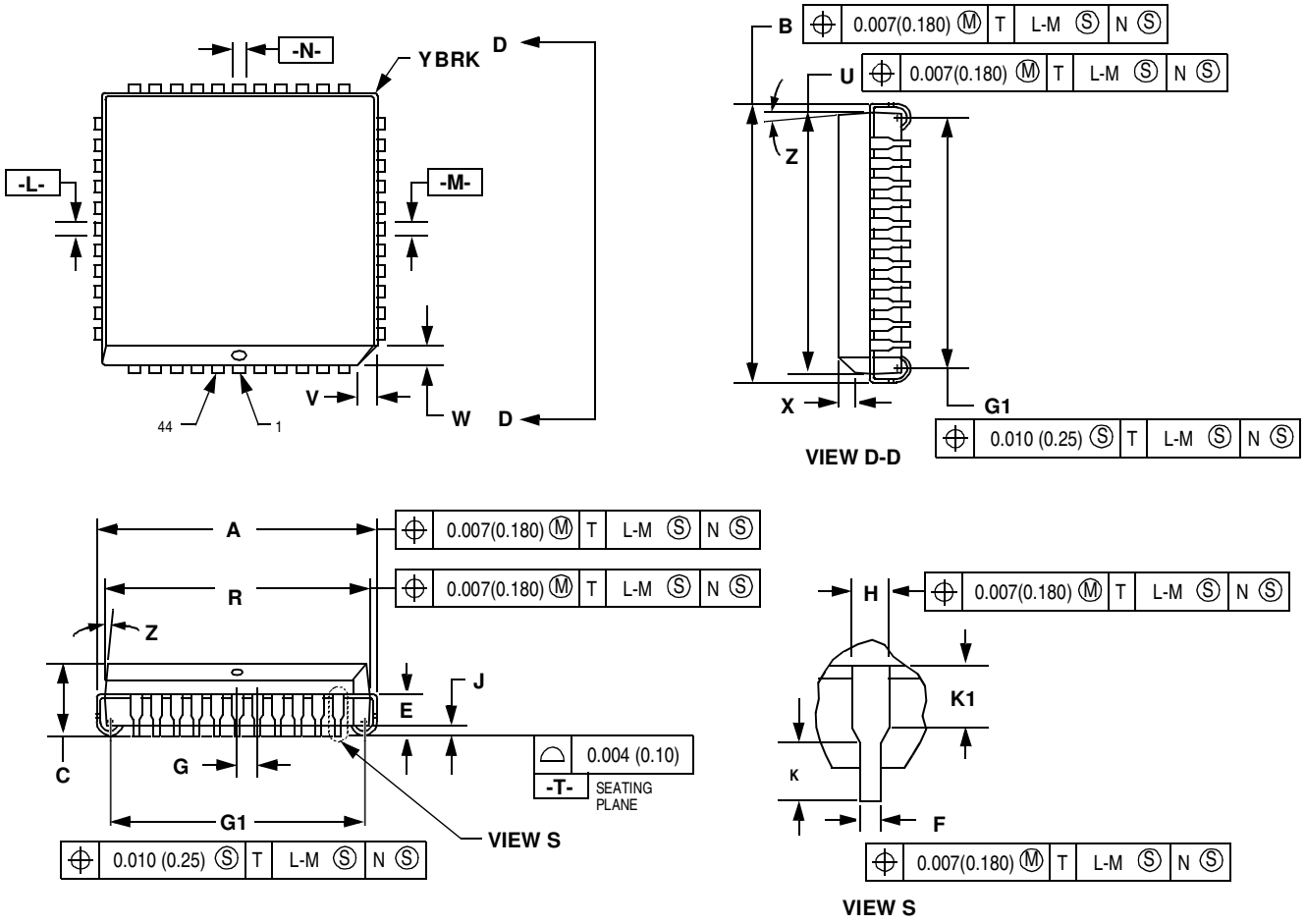
14.4 40-Pin Ceramic Dual In-Line Package (Cerdip)



DIM	INCHES		MILLIMETERS	
	MIN	MAX	MIN	MAX
A	2.020	2.096	51.31	53.23
B	0.500	0.610	12.70	15.94
C	0.160	0.240	4.06	6.09
D	0.015	0.022	0.38	0.55
F	0.050	0.065	1.27	1.65
G	0.100 BSC		2.54 BSC	
J	0.008	0.012	0.20	0.30
K	0.125	0.160	3.17	4.06
L	0.600 BSC		15.24 BSC	
M	0°	15°	0°	15°
N	0.020	0.050	0.51	1.27

Figure 14-2. MC68HC705C8AS Package Dimensions (Case #734A)

14.5 44-Lead Plastic-Leaded Chip Carrier (PLCC)

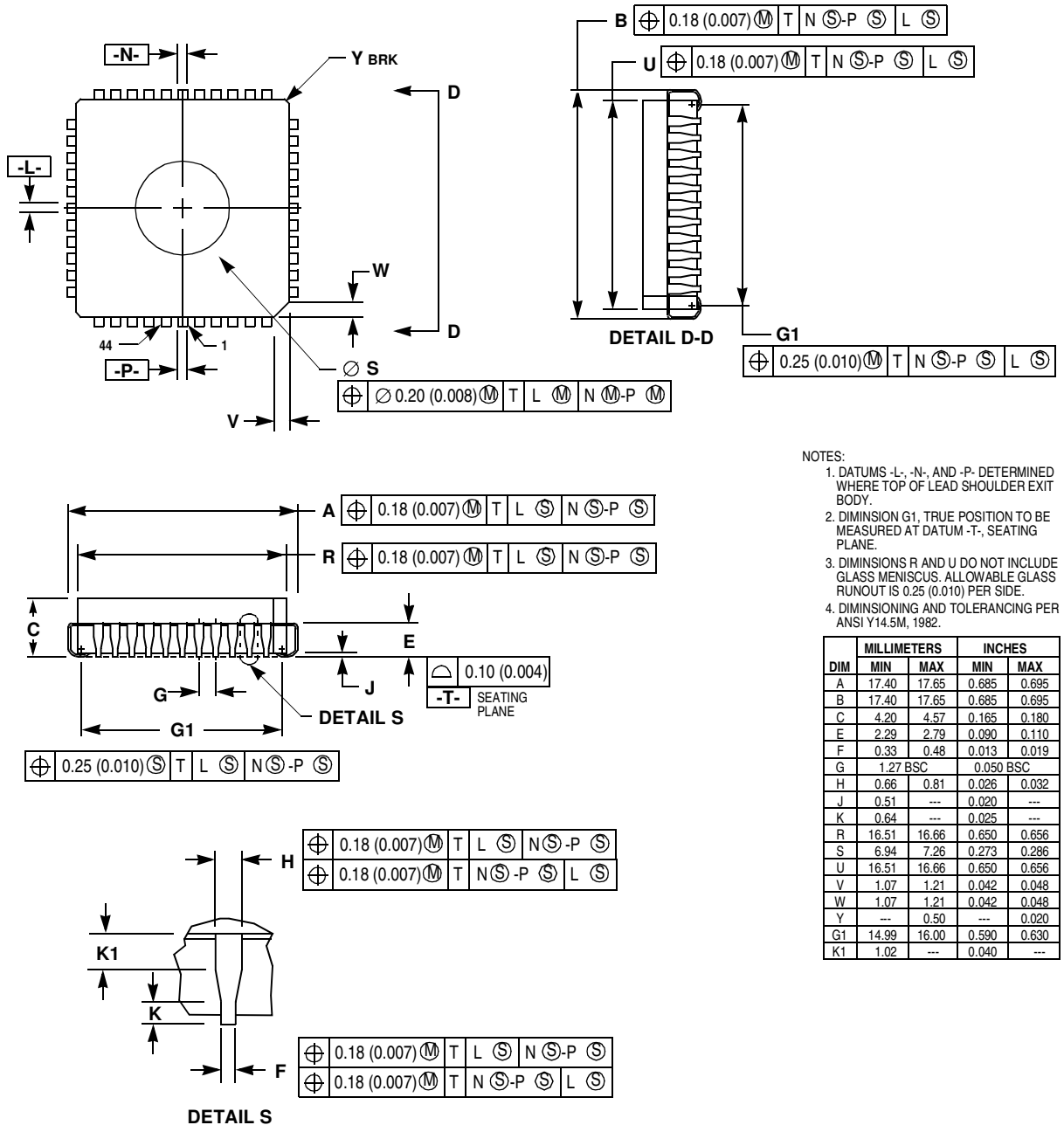


NOTES:
1. DATUMS -L-, -M-, AND -N- ARE DETERMINED WHERE TOP OF LEAD SHOLDERS EXITS PLASTIC BODY AT MOLD PARTING LINE.
2. DIMENSION G1, TRUE POSITION TO BE MEASURED AT DATUM -T-, SEATING PLANE.
3. DIMENSION R AND U DO NOT INCLUDE MOLD FLASH. ALLOWABLE MOLD FLASH IS 0.010 (0.25) PER SIDE.
4. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
5. CONTROLLING DIMENSION: INCH.
6. THE PACKAGE TOP MAY BE SMALLER THAN THE PACKAGE BOTTOM BY UP TO 0.012 (0.300). DIMENSIONS R AND U ARE DETERMINED AT THE OUTERMOST EXTREMES OF THE PLASTIC BODY EXCLUSIVE OF THE MOLD FLASH, TIE BAR BURRS, GATE BURRS AND INTERLEAD FLASH, BUT INCLUDING ANY MISMATCH BETWEEN THE TOP AND BOTTOM OF THE PLASTIC BODY.
7. DIMINSION H DOES NOT INCLUDE DAMBAR PROTRUSION OR INTRUSION. THE DAMBAR PROTUSION(S) SHALL NOT CAUSE THE H DIMINSION TO BE GREATER THAN 0.037 (0.940198). THE DAMBAR INTRUSION(S) SHALL NOT CAUSE THE H DIMINISION TO SMALLER THAN 0.025 (0.635).

	INCHES		MILLIMETERS	
DIM	MIN	MAX	MIN	MAX
A	0.685	0.695	17.40	17.65
B	0.685	0.695	17.40	17.65
C	0.165	0.180	4.20	4.57
E	0.090	0.110	2.29	2.79
F	0.013	0.019	0.33	0.48
G	0.050 BSC		1.27 BSC	
H	0.026	0.032	0.66	0.81
J	0.020	—	0.51	—
K	0.025	—	0.64	—
R	0.650	0.656	16.51	16.66
U	0.650	0.656	16.51	16.66
V	0.042	0.048	1.07	1.21
W	0.042	0.048	1.07	1.21
X	0.042	0.056	1.07	1.42
Y	—	0.020	—	0.50
Z	2°	10°	2°	10°
G1	0.610	0.630	15.50	16.00
K1	0.040	—	1.02	—

Figure 14-3. MC68HC705C8AFN Package Dimensions (Case #777)

14.6 44-Lead Ceramic-Leaded Chip Carrier (CLCC)

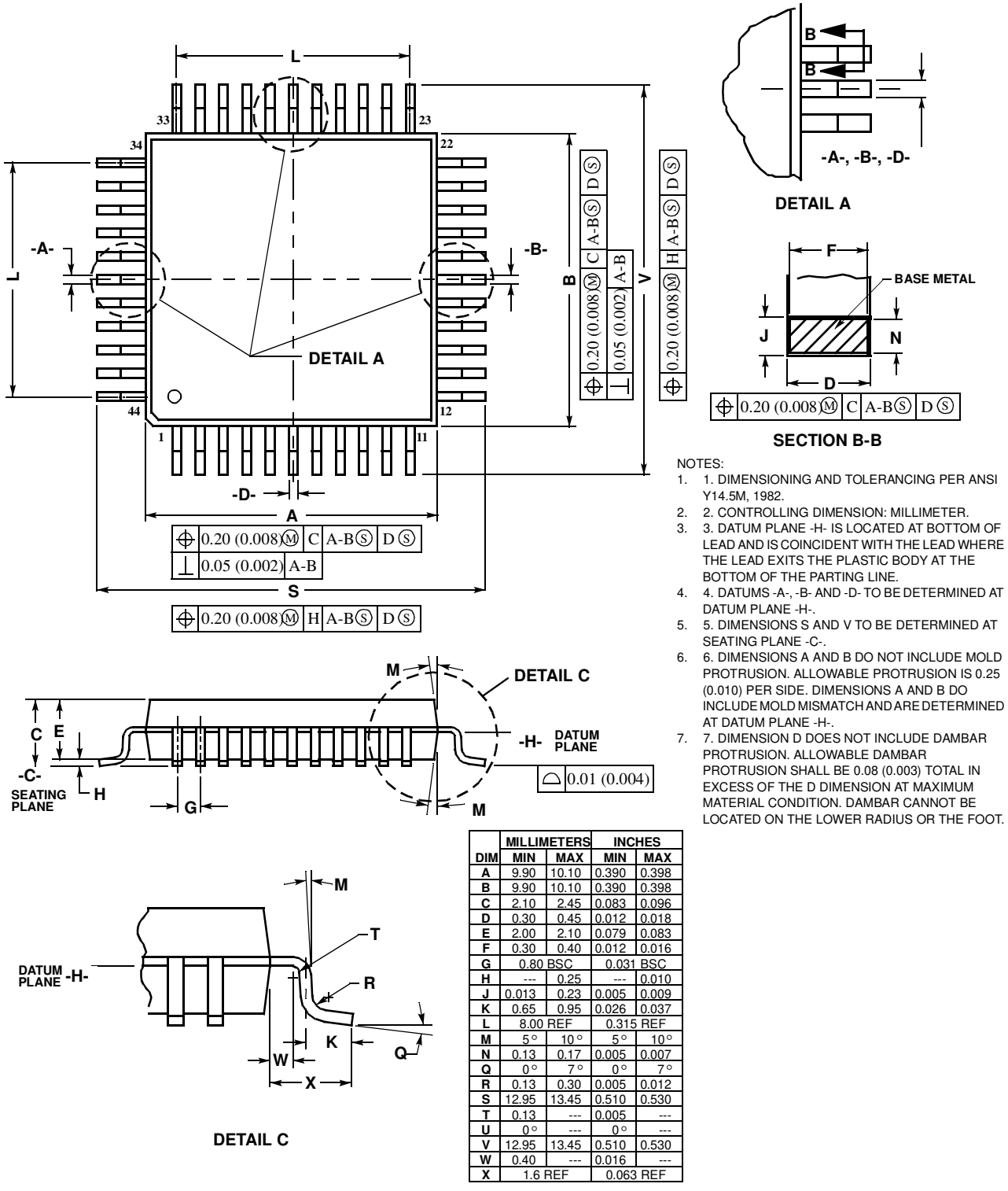


NOTES:
1. DATUMS -L-, -N-, AND -P- DETERMINED WHERE TOP OF LEAD SHOULDER EXIT BODY.
2. DIMINSION G1, TRUE POSITION TO BE MEASURED AT DATUM -T-, SEATING PLANE.
3. DIMINSIONS R AND U DO NOT INCLUDE GLASS MENISCUS. ALLOWABLE GLASS RUNOUT IS 0.25 (0.010) PER SIDE.
4. DIMINSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	17.40	17.65	0.685	0.695
B	17.40	17.65	0.685	0.695
C	4.20	4.57	0.165	0.180
E	2.29	2.79	0.090	0.110
F	0.33	0.48	0.013	0.019
G	1.27 BSC		0.050 BSC	
H	0.66	0.81	0.026	0.032
J	0.51	---	0.020	---
K	0.64	---	0.025	---
R	16.51	16.66	0.650	0.656
S	6.94	7.26	0.273	0.286
U	16.51	16.66	0.650	0.656
V	1.07	1.21	0.042	0.048
W	1.07	1.21	0.042	0.048
Y	---	0.50	---	0.020
G1	14.99	16.00	0.590	0.630
K1	1.02	---	0.040	---

Figure 14-4. MC68HC705C8AFS Package Dimensions (Case #777B)

14.7 44-Pin Quad Flat Pack (QFP)

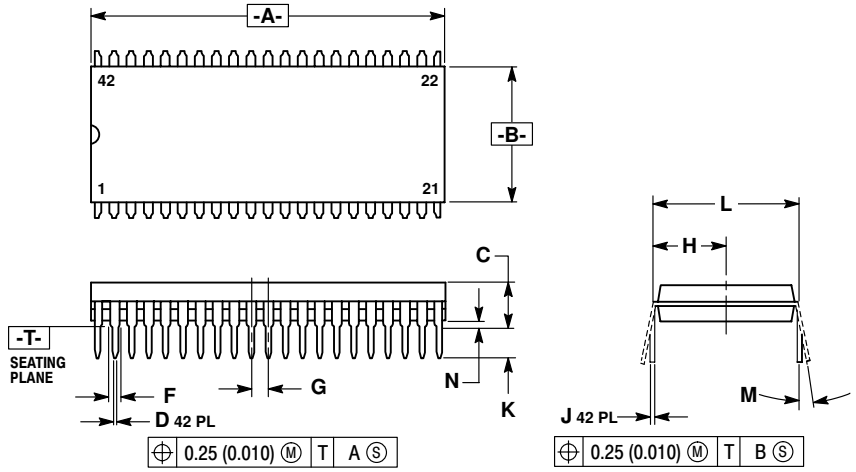


NOTES:
1. 1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. 2. CONTROLLING DIMENSION: MILLIMETER.
3. 3. DATUM PLANE -H- IS LOCATED AT BOTTOM OF LEAD AND IS COINCIDENT WITH THE LEAD WHERE THE LEAD EXITS THE PLASTIC BODY AT THE BOTTOM OF THE PARTING LINE.
4. 4. DATUMS -A-, -B- AND -D- TO BE DETERMINED AT DATUM PLANE -H-.
5. 5. DIMENSIONS S AND V TO BE DETERMINED AT SEATING PLANE -C-.
6. 6. DIMENSIONS A AND B DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE PROTRUSION IS 0.25 (0.010) PER SIDE. DIMENSIONS A AND B DO INCLUDE MOLD MISMATCH AND ARE DETERMINED AT DATUM PLANE -H-.
7. 7. DIMENSION D DOES NOT INCLUDE DAMBAR PROTRUSION. ALLOWABLE DAMBAR PROTRUSION SHALL BE 0.08 (0.003) TOTAL IN EXCESS OF THE D DIMENSION AT MAXIMUM MATERIAL CONDITION. DAMBAR CANNOT BE LOCATED ON THE LOWER RADIUS OR THE FOOT.

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	9.90	10.10	0.390	0.398
B	9.90	10.10	0.390	0.398
C	2.10	2.45	0.083	0.096
D	0.30	0.45	0.012	0.018
E	2.00	2.10	0.079	0.083
F	0.30	0.40	0.012	0.016
G	0.80 BSC		0.031 BSC	
H	---	0.25	---	0.010
J	0.013	0.23	0.005	0.009
K	0.65	0.95	0.026	0.037
L	8.00 REF		0.315 REF	
M	5°	10°	5°	10°
N	0.13	0.17	0.005	0.007
Q	0°	7°	0°	7°
R	0.13	0.30	0.005	0.012
S	12.95	13.45	0.510	0.530
T	0.13	---	0.005	---
U	0°	---	0°	---
V	12.95	13.45	0.510	0.530
W	0.40	---	0.016	---
X	1.6 REF		0.063 REF	

Figure 14-5. MC68HC705C8AFB Package Dimensions (Case #824A)

14.8 42-Pin Shrink Dual In-Line Package (SDIP)



NOTES:
1. DIMENSIONS AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: INCH.
3. DIMENSION L TO CENTER OF LEAD WHEN FORMED PARALLEL.
4. DIMENSIONS A AND B DO NOT INCLUDE MOLD FLASH. MAXIMUM MOLD FLASH 0.25 (0.010).

	INCHES		MILLIMETERS	
DIM	MIN	MAX	MIN	MAX
A	1.435	1.465	36.45	37.21
B	0.540	0.560	13.72	14.22
C	0.155	0.200	3.94	5.08
D	0.014	0.022	0.36	0.56
F	0.032	0.046	0.81	1.17
G	0.070 BSC		1.778 BSC	
H	0.300 BSC		7.62 BSC	
J	0.008	0.015	0.20	0.38
K	0.115	0.135	2.92	3.43
L	0.600 BSC		15.24 BSC	
M	0°	15°	0°	15°
N	0.020	0.040	0.51	1.02

Figure 14-6. MC68HC705C8AB Package Dimensions (Case #858)

Mechanical Specifications

Section 15. Ordering Information

15.1 Contents

15.2 Introduction

This section contains ordering information for the available package types.

15.3 MCU Order Numbers

Table 15-1 lists the MC order numbers.

Table 15-1. MC68HC705C8A Order Numbers

Package Type	Temperature Range	Order Number
40-pin plastic dual in-line package (PDIP)	–40°C to +85°C	MC68HC705C8AC(1)P(2)
44-lead plastic-leaded chip carrier (PLCC)	–40°C to +85°C	MC68HC705C8ACFN(3)
44-lead ceramic-leaded chip carrier (CLCC)	–40°C to +85°C	MC68HC705C8ACFS(4)
40-pin windowed ceramic DIP (Cerdip)	–40°C to +85°C	MC68HC705C8ACS(5)
44-pin quad flat pack (QFP)	–40°C to +85°C	MC68HC705C8ACFB(6)
42-pin shrink dual in-line package (SDIP)	–40°C to +85°C	MC68HC705C8ACB(7)

1. C = Extended temperature range (–40°C to +85°C)
2. P = Plastic dual in-line package (PDIP)
3. FN = Plastic-leaded chip carrier (PLCC)
4. FS = Ceramic-leaded chip carrier (CLCC)
5. S = Windowed ceramic dual in-line package (Cerdip)
6. FB = Quad flat pack (QFP)
7. B = Shrink dual in-line package (SDIP)

Ordering Information

Appendix A. MC68HSC705C8A

A.1 Contents

A.2 Introduction

The MC68HSC705C8A is an enhanced, high-speed version of the MC68HC705C8A, featuring a 4-MHz bus speed.

The data in this document, *MC68HC705C8A Technical Data* Rev. 3, applies to the MC68HSC705C8A with the exceptions given in this appendix.

The computer operating properly (COP) mode bits (CM1 and CM0 in the COP control register) select the timeout period of the programmable COP watchdog, as shown in **Table A-1**. See **Figure 5-3. Programmable COP Control Register (COPCR)**.

Table A-1. Programmable COP Timeout Period Selection

CM1:CM0	COP Timeout Rate	Programmable COP Timeout Period			
		f_{OSC} = 8.0 MHz f_{OP} = 4.0 MHz	f_{OSC} = 4.0 MHz f_{OP} = 2.0 MHz	f_{OSC} = 3.5795 MHz f_{OP} = 1.7897 MHz	f_{OSC} = 2.0 MHz f_{OP} = 1.0 MHz
00	$f_{OP} \div 2^{15}$	8.192 ms	16.38 ms	18.31 ms	32.77 ms
01	$f_{OP} \div 2^{17}$	32.77 ms	65.54 ms	73.24 ms	131.07 ms
10	$f_{OP} \div 2^{19}$	131.07 ms	262.14 ms	292.95 ms	524.29 ms
11	$f_{OP} \div \div 2^{21}$	524.29 ms	1.048 s	1.172 s	2.097 s

A.3 5.0-Volt High-Speed DC Electrical Characteristics

Characteristic[1]	Symbol	Min	Typ[2]	Max	Unit
Output high voltage I_{Load} = –0.8 mA PA7–PA0, PB7–PB0, PC6–PC0, TCMP I_{Load} = –1.6 mA PD4–PD1 I_{Load} = –5.0 mA PC7	V_{OH}	V_{DD} – 0.8	— — —	— — —	V
Output low voltage I_{Load} = 1.6 mA PA7–PA0, PB7–PB0, PC6–PC0, PD4–PD1 I_{Load} = 20 mA PC7	V_{OL}	— —	— —	0.4 0.4	V
Supply current[3] Run[4] Wait[5] Stop[6] 25°C –40°C to +85°C	I_{DD}	— — — —	5.92 2.27 5 2.0	14 7.0 50 50	mA mA µA µA

1. V_{DD} = 5 V ± 10%; V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, unless otherwise noted
2. Typical values reflect average measurements at midpoint of voltage range at 25°C.
3. I_{DD} measured with port B pullup devices disabled.
4. Run (operating) I_{DD} measured using external square wave clock source (f_{OSC} = 8.0 MHz). All inputs 0.2 V from rail. No dc loads. Less than 50 pF on all outputs. C_L = 20 pF on OSC2. OSC2 capacitance linearly affects run I_{DD}.
5. Wait I_{DD} measured using external square wave clock source (f_{OSC} = 8.0 MHz). All inputs 0.2 V from rail. No dc loads. Less than 50 pF on all outputs. C_L = 20 pF on OSC2. V_{IL} = 0.2 V, $V_{IH} = V_{DD}$ – 0.2 V. All ports configured as inputs. SPI and SCI disabled. If SPI and SCI enabled, add 10% current draw. OSC2 capacitance linearly affects wait I_{DD}.
6. Stop I_{DD} measured with OSC1 = V_{DD}. All ports configured as inputs. V_{IL} = 0.2 V, $V_{IH} = V_{DD}$ – 0.2 V.

A.4 3.3-Volt High-Speed DC Electrical Characteristics

Characteristic[1]	Symbol	Min	Typ[2]	Max	Unit
Output high voltage I_{Load} = –0.2 mA PA7–PA0, PB7–PB0, PC6–PC0, TCMP I_{Load} = –0.4 mA PD4–PD1 I_{Load} = –1.5 mA PC7	V_{OH}	V_{DD} – 0.3	— — —	— — —	V
Output low voltage I_{Load} = 0.4 mA PA7–PA0, PB7–PB0, PC6–PC0, PD4–PD1 I_{Load} = 6.0 mA PC7	V_{OL}	— —	— —	0.3 0.3	V
Supply current[3] Run[4] Wait[5] Stop[6]	I_{DD}	— — —	1.91 0.915 2.0	6.0 2.0 20	mA mA µA

1. V_{DD} = 3.3 V ± 10%; V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, unless otherwise noted
2. Typical values reflect average measurements at midpoint of voltage range at 25°C.
3. I_{DD} measured with port B pullup devices disabled.
4. Run (operating) I_{DD} measured using external square wave clock source (f_{OSC} = 4.2 MHz). All inputs 0.2 V from rail. No dc loads. Less than 50 pF on all outputs. C_L = 20 pF on OSC2. OSC2 capacitance linearly affects run I_{DD}.
5. Wait I_{DD} measured using external square wave clock source (f_{OSC} = 4.2 MHz). All inputs 0.2 V from rail. No dc loads. Less than 50 pF on all outputs. C_L = 20 pF on OSC2. V_{IL} = 0.2 V, $V_{IH} = V_{DD} – 0.2$ V. All ports configured as inputs. SPI and SCI disabled. If SPI and SCI enabled, add 10% current draw. OSC2 capacitance linearly affects wait I_{DD}.
6. Stop I_{DD} measured with OSC1 = V_{DD}. All ports configured as inputs. V_{IL} = 0.2 V; $V_{IH} = V_{DD} – 0.2$ V.

A.5 5.0-Volt High-Speed Control Timing

Characteristic[1]	Symbol	Min	Max	Unit
Oscillator frequency Crystal oscillator External clock	f_{OSC}	— dc	8.0 8.0	MHz
Internal operating frequency ($f_{OSC} \div 2$) Crystal oscillator External clock	f_{OP}	— dc	4.0 4.0	MHz
Cycle time	t_{CYC}	250	—	ns
Input capture pulse width	t_{TH}, t_{TL}	65	—	ns
Interrupt pulse width low (edge-triggered)	t_{ILIH}	65	—	ns
OSC1 pulse width	t_{OH}, t_{OL}	45	—	ns

1. V_{DD} = 5 V ± 10%; V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, unless otherwise noted

A.6 3.3-Volt High-Speed Control Timing

Characteristic[1]	Symbol	Min	Max	Unit
Oscillator frequency Crystal oscillator External clock	f_{OSC}	— dc	4.0 4.0	MHz
Internal operating frequency ($f_{OSC} \div 2$) Crystal oscillator External clock	f_{OP}	— dc	2.0 2.0	MHz
Cycle time	t_{CYC}	476	—	ns
Input capture pulse width	t_{TH}, t_{TL}	125	—	ns
Interrupt pulse width low (edge-triggered)	t_{ILIH}	125	—	ns
OSC1 pulse width	t_{OH}, t_{OL}	90	—	ns

1. V_{DD} = 3.3 V ± 10%; V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, unless otherwise noted

A.7 5.0-Volt High-Speed SPI Timing

Diagram Number[1]	Characteristic[2]	Symbol	Min	Max	Unit
	Operating frequency Master Slave	$f_{OP(S)}$ $f_{OP(S)}$	dc dc	0.5 4.0	f_{OP} MHz
1	Cycle time Master Slave	$t_{CYC(M)}$ $t_{CYC(S)}$	2.0 250	— —	t_{CYC} ns
2	Enable lead time Master Slave	$t_{Lead(M)}$ $t_{Lead(S)}$	Note [3] 125	— —	ns
3	Enable lag time Master Slave	$t_{Lag(M)}$ $t_{Lag(S)}$	Note[2] 375	— —	ns
4	Clock (SCK) high time Master Slave	$t_{W(SCKH)M}$ $t_{W(SCKH)S}$	170 95	— —	ns
5	Clock (SCK) low time Master Slave	$t_{W(SCKL)M}$ $t_{W(SCKL)S}$	170 95	— —	ns
6	Data setup time (inputs) Master Slave	$t_{SU(M)}$ $t_{SU(S)}$	50 50	— —	ns
7	Data hold time (inputs) Master Slave	$t_{H(M)}$ $t_{H(S)}$	50 50	— —	ns
8	Access time[4] Slave	t_A	0	60	ns
9	Disable time[5] Slave	t_{DIS}	—	120	ns
10	Data valid time Master (before capture edge) Slave (after enable edge)[6]	$t_{V(M)}$ $t_{V(S)}$	0.25 —	— 120	$t_{CYC(M)}$ ns

Continued

Diagram Number[1]	Characteristic[2]	Symbol	Min	Max	Unit
11	Data hold time (outputs) Master (after capture edge) Slave (after enable edge)	 $t_{HO(M)}$ $t_{HO(S)}$	 0.25 0	 — —	$t_{CYC(M)}$ ns
12	Rise time[7] SPI outputs (SCK, MOSI, MISO) SPI inputs (SCK, MOSI, MISO, $\overline{SS}$)	 t_{RM} t_{RS}	 — —	 50 2.0	 ns µs
13	Fall time[8] SPI outputs (SCK, MOSI, MISO) SPI inputs (SCK, MOSI, MISO, $\overline{SS}$)	 t_{FM} t_{FS}	 — —	 50 2.0	 ns µs

1. Diagram numbers refer to dimensions in **Figure 13-8. SPI Master Timing** and **Figure 13-9. SPI Slave Timing**.
2. V_{DD} = 5 V ± 10%; V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, unless otherwise noted
3. Signal production depends on software.
4. Time to data active from high-impedance state
5. Hold time to high-impedance state
6. With 200 pF on all SPI pins.
7. 20% of V_{DD} to 70% of V_{DD}; C_L = 200 pF
8. 70% of V_{DD} to 20% of V_{DD}; C_L = 200 pF

A.8 3.3-Volt High-Speed SPI Timing

Diagram Number[1]	Characteristic[2]	Symbol	Min	Max	Unit
	Operating frequency Master Slave	$f_{OP(S)}$ $f_{OP(S)}$	dc dc	0.5 2.1	f_{OP} MHz
1	Cycle time Master Slave	$t_{CYC(M)}$ $t_{CYC(S)}$	2.0 480	— —	t_{CYC} ns
2	Enable lead time Master Slave	$t_{Lead(M)}$ $t_{Lead(S)}$	Note [3] 240	— —	ns
3	Enable lag time Master Slave	$t_{Lag(M)}$ $t_{Lag(S)}$	Note[2] 720	— —	ns
4	Clock (SCK) high time Master Slave	$t_{W(SCKH)M}$ $t_{W(SCKH)S}$	340 190	— —	ns
5	Clock (SCK) low time Master Slave	$t_{W(SCKL)M}$ $t_{W(SCKL)S}$	340 190	— —	ns
6	Data setup time (inputs) Master Slave	$t_{SU(M)}$ $t_{SU(S)}$	100 100	— —	ns
7	Data hold time (inputs) Master Slave	$t_{H(M)}$ $t_{H(S)}$	100 100	— —	ns
8	Access time[4] Slave	t_A	0	120	ns
9	Disable time[5] Slave	t_{DIS}	—	240	ns
10	Data valid time Master (before capture edge) Slave (after enable edge)[6]	$t_{V(M)}$ $t_{V(S)}$	0.25 —	— 240	$t_{CYC(M)}$ ns

Continued

Diagram Number[1]	Characteristic[2]	Symbol	Min	Max	Unit
11	Data hold time (outputs) Master (after capture edge) Slave (after enable edge)	 $t_{HO(M)}$ $t_{HO(S)}$	 0.25 0	 — —	 $t_{CYC(M)}$ ns
12	Rise time[7] SPI outputs (SCK, MOSI, MISO) SPI inputs (SCK, MOSI, MISO, $\overline{SS}$)	 t_{RM} t_{RS}	 — —	 100 2.0	 ns µs
13	Fall time[8] SPI outputs (SCK, MOSI, MISO) SPI inputs (SCK, MOSI, MISO, $\overline{SS}$)	 t_{FM} t_{FS}	 — —	 100 2.0	 ns µs

1. Diagram numbers refer to dimensions in **Figure 13-8. SPI Master Timing** and **Figure 13-9. SPI Slave Timing**.
2. V_{DD} = 3.3 V ± 10%; V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, unless otherwise noted
3. Signal production depends on software.
4. Time to data active from high-impedance state
5. Hold time to high-impedance state
6. With 200 pF on all SPI pins
7. 20% of V_{DD} to 70% of V_{DD}; C_L = 200 pF
8. 70% of V_{DD} to 20% of V_{DD}; C_L = 200 pF

A.9 Ordering Information

Table A-2 provides ordering information for the MC68HSC705C8A.

Table A-2. MC68HSC705C8A Order Numbers

Package Type	Temperature Range	Order Number
40-pin plastic dual in-line package (PDIP)	–40°C to +85°C	MC68HSC705C8AC(1)P(2)
44-lead plastic-leaded chip carrier (PLCC)	–40°C to +85°C	MC68HSC705C8ACFN(3)
44-lead ceramic-leaded chip carrier (CLCC)	–40°C to +85°C	MC68HSC705C8ACFS(4)
40-pin ceramic DIP (cerdip)	–40°C to +85°C	MC68HSC705C8ACS(5)
44-pin quad flat pack (QFP)	–40°C to +85°C	MC68HSC705C8ACFB(6)
42-pin shrink dual in-line package (SDIP)	–40°C to +85°C	MC68HSC705C8ACB(7)

1. C = Extended temperature range (–40°C to +85°C)
2. P = Plastic dual in-line package (PDIP)
3. FN = Plastic-leaded chip carrier (PLCC)
4. FS = Ceramic-leaded chip carrier (CLCC)
5. S = Windowed ceramic dual in-line package (cerdip)
6. FB = Quad flat pack (QFP)
7. B = Shrink dual in-line package (SDIP)

Index

D

E

F

H

I

J

L

M

O

P

R

S

T

V

W

Index

HOW TO REACH US:

USA/EUROPE/LOCATIONS NOT LISTED:

Motorola Literature Distribution;
P.O. Box 5405, Denver, Colorado 80217
1-303-675-2140 or 1-800-441-2447

JAPAN:

Motorola Japan Ltd.; SPS, Technical Information Center,
3-20-1, Minami-Azabu Minato-ku, Tokyo 106-8573 Japan
81-3-3440-3569

ASIA/PACIFIC:

Motorola Semiconductors H.K. Ltd.;
Silicon Harbour Centre, 2 Dai King Street,
Tai Po Industrial Estate, Tai Po, N.T., Hong Kong
852-26668334

TECHNICAL INFORMATION CENTER:

1-800-521-6274

HOME PAGE:

http://www.motorola.com/semiconductors

MC68HC705C8A/D

68HC705C8A : Microcontroller

The 68HC05C8A includes keyboard scanning logic, a high current pin, a COP watchdog timer, and read-only (ROM) security feature.

68HC705C8A Features

- 68HC05 CPU
  - 8Kbytes EPROM
  - 304 bytes RAM
- Timer System
  - 16-bit Capture/Compare Timer system
  - Computer Operating Properly Watchdog Timer
- 24 Bidirectional I/O Lines and 7 Input-Only lines
- On-Chip Oscillator with Crystal/Ceramic Resonator
- Fully Static Operation
- Serial Communications
  - Asynchronous Serial Communications Interface (SCI)
  - Synchronous Serial Peripheral Interface (SPI)

[top]

68HC705C8A Parametrics

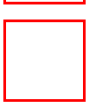
RAM (Bytes)	EPROM/OTP (Bytes)	Timer	I/O	Serial	Operating Voltage (V)	Bus Frequency (Max) (MHz)	Availability
304	8K	16-Bit, 1 I/C, 1 O/C	31	SCI, SPI	3.3, 5.0	4.0	Now

[top]

68HC705C8A Documentation

Application Note

ID	Name	Format	Size K	Rev #	Date Last Modified	Order Availability
AN-HK-22/D	MC68HC05SR3 and MC68HC705SR3 Design Notes	pdf	3241	0	1/01/1994	-
AN-HK-23/D	MC6805R3 and MC68HC05SR3 Technical Comparison	pdf	544	0	1/01/1994	-

AN1050/D	Designing for Electromagnetic Compatibility (EMC) with HCMOS Microcontrollers	pdf	82	0	1/01/2000	
AN1055/D	M6805 16-Bit Support Macros	pdf	1048	0	1/01/1990	
AN1066/D	Interfacing the MC68HC05C5 SIOP to an I2C Peripheral	pdf	196	1	11/01/2001	
AN1067/D	Pulse Generation and Detection with Microcontroller Units	pdf	242	1	5/31/2002	
AN1212/D	J1850 Multiplex Bus Communication Using the MC68HC705C8 and the SC371016 J1850 Communications Interface (JCI)	pdf	377	1	11/13/2001	
AN1212SW	Software files for AN1212 zipped (exe file included)	zip	3329	0	1/01/1992	-
AN1222/D	Arithmetic Waveform Synthesis with the HC05/08 MCUs	pdf	24	0	1/01/1993	
AN1222SW	Software Files for AN1222 zipped	zip	20	0	1/01/1995	-
AN1226/D	Use of the 68HC705C8A in Place of a 68HC705C8	pdf	90	4	1/01/1996	
AN1227/D	Using 9346 Series Serial EEPROMs with 6805 Series Microcontrollers	pdf	401	1	1/01/1997	
AN1227SW	Software files zipped for AN1227	zip	151	1	1/01/1997	-
AN1228/D	Interfacing the M68HC05 MCU to the MC145051 A/D Converter	pdf	160	2	1/01/1997	
AN1228SW	Software for AN1228 zippedr	zip	27	2	1/01/1997	-
AN1256/D	Interfacing the HC05 MCU to a Multichannel Digital-to-Analog Converter Using the MC68HC705C8A and the MC68HC705J1A	pdf	89	1.1	1/01/1995	
AN1256SW	Software files for AN1256 zipped	zip	80	1.1	1/01/1995	-
AN1259/D	System Design and Layout Techniques for Noise Reduction in MCU-Based Systems	pdf	78	0	1/01/1995	
AN1262/D	Simple Real-Time Kernels for M68HC05 Microcontrollers	pdf	84	0	1/01/1995	
AN1262SW	Software files for AN1262	zip	11	0	1/01/1995	-
AN1263/D	Designing for Electromagnetic Compatibility with Single-Chip Microcontrollers	pdf	104	0	1/01/1995	
AN1292/D	Adding a Voice User Interface to M68HC05 Applications	pdf	155	0	1/01/1996	
AN1292SW	Software files for AN1292 zipped	zip	215	0	1/01/1996	-
AN1298/D	Variations in the Motorola MC68HC(7)05Cx Family	pdf	200	0	1/01/1996	
AN1667/D	Software SCI Implementation to the MISC Communication Protocol	pdf	112	0	7/10/2002	
AN1667SW	Software for AN1667, zip format	zip	93	1.0	7/31/2002	-
AN1688/D	MISC Bus Slave Switch Node	pdf	243	0	7/11/2002	

AN1705/D	Noise Reduction Techniques for Microcontroller-Based Systems	pdf	67	0	1/01/1999	
AN1723/D	Interfacing MC68HC05 Microcontrollers to the IBM AT Keyboard Interface	pdf	274	0	1/01/1997	
AN1734/D	Pulse Width Modulation Using the 16-Bit Timer	pdf	102	0	1/01/1998	
AN1734SW	Software files for AN1734 zipped	zip	2	0	1/01/1997	-
AN1744/D	Resetting Microcontrollers During Power Transitions	pdf	80	0	1/01/1998	
AN1745/D	Interfacing the HC705C8A to an LCD Module	pdf	157	0	1/01/1998	
AN1745SW	Software files for AN1745 zipped	zip	3	0	1/01/1998	-
AN1752/D	Data Structures for 8-Bit Microcontrollers	pdf	213	1	5/07/2001	
AN1755/D	Interfacing the MC68HC705C8A to the DS2430A 256-Bit 1-Wire EEPROM	pdf	129	0	1/01/1998	
AN1757/D	Add a Unique Silicon Serial Number to the HC05	pdf	105	0	1/01/1998	
AN1758/D	Add Addressable Switches to the HC05	pdf	111	0	1/01/1998	
AN1761/D	Interfacing the MC68HC705C8A to the X76F041 PASS SecureFlash	pdf	179	0	1/01/1998	
AN1771/D	Precision Sine-Wave Tone Synthesis Using 8-Bit MCUs	pdf	250	0	1/01/1998	
AN1775/D	Expanding Digital Input with an A/D Converter	pdf	86	1	1/01/1998	
AN1818/D	Software SCI Routines with the 16-Bit Timer Module	pdf	84	0	1/01/1999	
AN1820/D	Software I2C Communications	pdf	55	0	1/01/1999	
AN1820SW	Software files for AN1820 zipped	zip	2	0	1/01/1998	-
AN2103/D	Local Interconnect Network (LIN) Demonstration	pdf	953	0	12/01/2000	
AN2159/D	Digital Direct Current Ignition System Using HC08 Microcontrollers	pdf	129	0	11/20/2001	
AN2159SW	AN2159SW	zip	182	1	3/08/2002	-
AN4006/D	Digital Captive Discharge Ignition System Using HC05/HC08 8-Bit Microcontrollers	pdf	61	0	3/27/2000	
AN442/D	Driving LCDs with M6805 Microprocessors	pdf	1134	0	1/01/1991	
AN463/D	68HC05K0 Infra-red Remote Control	pdf	111	0	1/01/1992	
AN464/D	Software Driver Routines for the Motorola MC68HC05 CAN Module	pdf	2859	0	1/01/1993	

AN477/D	Simple A/D for MCUs without Built-In A/D Converters	pdf	224	0	1/01/1993	
AN499/D	Let the MC68HC705 Program Itself	pdf	154	0	7/01/1996	
AN991/D	Using the Serial Peripheral Interface to Communicate Between Multiple Microcomputers	pdf	251	1	1/28/2002	
ANE416/D	MC68HC05B4 Radio Synthesizer	pdf	1958	0	1/01/1988	

Brochure

ID	Name	Format	Size K	Rev #	Date Last Modified	Order Availability
FLYREMBEDFLASH/D	Embedded Flash: Changing the Technology World for the Better	pdf	621	1	4/03/2002	

Data Sheets

ID	Name	Format	Size K	Rev #	Date Last Modified	Order Availability
MC68HC705C8A/D	68HC705C8A Technical Data Book	pdf	2428	3	3/21/2002	

Engineering Bulletin

ID	Name	Format	Size K	Rev #	Date Last Modified	Order Availability
EB166/D	System Design Considerations: Converting from the MC68HC805B6 to the MC68HC705B16 Microcontroller	pdf	757	0	1/01/1993	
EB180/D	Differences between the MC68HC705B16 and the MC68HC705B16N	pdf	20	0	1/01/1996	
EB181/D	Frequently Asked Questions and Answers for the M68HC05 Family MCAN Module	pdf	181	0	1/01/1997	
EB349/D	RAM Data Retention Considerations for Motorola Microcontrollers	pdf	45	1	6/22/2000	
EB396/D	Use of OSC2/XTAL as a Clock Output on Motorola Microcontrollers	pdf	49	0	6/19/2002	
EB413/D	Resetting MCUs	pdf	62	0	1/01/2000	
EB421/D	The Motorola MCAN Module	pdf	78	0	2/23/2000	

Errata

ID	Name	Format	Size K	Rev #	Date Last Modified	Order Availability
68HC705C8AMSE1/D	68HC705C8A Device Information Sheet: 0E20T-2E79R Mask Sets	pdf	11	1	11/06/1996	-

68HC705C8AMSE2/D	68HC705C8A Device Information Sheet: 1E20T-3E79R Mask Sets	pdf	14	1	1/01/1995	-
68HC705C8AMSE3/D	68HC705C8A Device Information Sheet: 0E20T-3E79R Mask Sets	pdf	17	3	7/16/2001	-

Product Change Notices

ID	Name	Format	Size K	Rev #	Date Last Modified	Order Availability
PCN7701	QFP 10X10 ASSY MOVE FROM SHC TO BAT3	htm	16	-	7/09/2002	-
PCN7899	44/52/68 PLCC ASSY MOVE FROM SDI TO KLM	htm	31	0	8/14/2002	-
PCN7900	LQFP 10X10 ASSY MOVE FROM SDI TO BAT3	htm	16	0	8/14/2002	-
PCN7977	14X14 QFP ASSY MOVE FROM SDI TO KLM	htm	17	0	9/12/2002	-
PCN8103	10X10 LQFP ASSY MOVE FROM SHC TO BAT3	htm	16	0	10/08/2002	-

Reference Manual

ID	Name	Format	Size K	Rev #	Date Last Modified	Order Availability
M68HC05AG/AD	M68HC05 Applications Guide	pdf	3272	4	3/18/2002	-
M68HC05TB/D	HC05 Family - Understanding Small Microcontrollers	pdf	2866	2	1/01/1998	
MC68HC05CXRG/D	MC68HC05C4, C8, C9, MC68HC705C8, MC68HC805C4, MC68HCL05C4, C8, MC68HSC05C4, C8 Programming Reference	pdf	3150	1	2/23/2000	-

Selector Guide

ID	Name	Format	Size K	Rev #	Date Last Modified	Order Availability
SG1006/D	Microcontrollers SPS Sales Guide	pdf	600	0	9/26/2002	
SG1011/D	Software and Development Tools Sales Guide	pdf	259	1	9/26/2002	
SG2000CR/D	Application Selector Guide Index and Cross-Reference.	pdf	62	0	6/24/2002	

[top]

68HC705C8A Development Tools/Boards

ID	Name	Vendor ID	Order Availability
M68ICS05C	In-Circuit Simulator for the 68HC05C and 68HC705C	MOTOROLA	
M68EM05C9A	Emulation Module	MOTOROLA	

KITMMDS05C	Modular Development System (MMDS) Kits	MOTOROLA	
KITMMEVS05C	Modular Evaluation System (MMEVS)	MOTOROLA	
CWHC05	CodeWarrior Development Tools for HC05	METROWERKS	

[top]

Design Tools

Software

ID	Name	Vendor ID	Format	Size K	Rev #
HC705C8AH	C header file for 68HC705C8A	MOTOROLA	zip	10	-
MATH16ACOD	General Math routines		asm	4	-
MATHAGBCOD	General Math routines		asm	6	-

Software Tools/Assemblers

ID	Name	Vendor ID	Format	Size K	Rev #
ASHC5ASM	DOS based freeware assembler	MOTOROLA	arc	55	0

Software/Application Software/Code Examples

ID	Name	Vendor ID	Format	Size K	Rev #
C8THERMSW	HC05 Software Example: Home Thermostat example using the 705C8 with indoor/outdoor temperature and time of day	MOTOROLA	zip	11	-
FLOAT05COD	Floating Point routines	MOTOROLA	zip	13	-
HC05DELAYSW	HC05 Software Example: Subroutine that delays for a whole number of milliseconds	MOTOROLA	zip	2	-
HC05EXSW	Library containing software examples in assembly for 68HC05	MOTOROLA	zip	45	-
HC05KEYINTSW	HC05 Software Examples: Using keyboard interrupts and decoding a matrix keypad	MOTOROLA	zip	7	-
HC05KEYPADSW	HC05 Software Example: Keypad debounce and decode. When a key is found, it is changed to ASCII and displayed on an LCD	MOTOROLA	zip	2	-
HC05LCDSW	HC05 Software Example: Initializes an LCD and displays ABCDEF...S	MOTOROLA	zip	1	-
HC05SCISW	HC05 Software Example: Serial Communications Interface example	MOTOROLA	zip	1	-
HC05SPISW	HC05 Software Example: Serial Peripheral Interface example	MOTOROLA	zip	1	-
HC05SWITCHSW	HC05 Software Example: Simple program that reads the state of a switch on a general-purpose I/O pin and lights an LED based on the state of the switch	MOTOROLA	zip	1	-

HC05TIMERSW	HC05 Software Example: Using the 68HC05 16-bit Timer	MOTOROLA	zip	1	-
J1APWMSW	HC05 Software Example: Low frequency PWM example using the 68HC705J1A real-time interrupt and timer overflow interrupt	MOTOROLA	zip	1	-
J1AUARTSW	HC05 Software example: Software UART example that transmits and receives data on the 68HC705J1A	MOTOROLA	zip	2	-
K1THERMSW	HC05 Software Example: Thermometer project using the 68HC705K1	MOTOROLA	zip	5	-
SAMPPROGCOD	Example routines	MOTOROLA	exe	35	-
THERM-CCOD	Thermometer example in C	MOTOROLA	zip	11	-

Software/Operating Systems

ID	Name	Vendor ID	Format	Size K	Rev #
PE68HC05SIM	Windows upgrades for P&E's simulator software for 68HC05	PEMICRO	html	0	-

[top]

Orderable Parts Information

PartNumber	Package Info	Life Cycle Description (code)	Remarks	Budgetary Price QTY 1000+ ($US)	Order Availability
MC68HC705C8ACP	40-Pin Plastic Dual In-Line Package (PDIP)	PRODUCT MATURITY/SATURATION(4)	-40 to +85 C	$3.95	
MC68HC705C8ACFN	44-Lead Plastic Leaded Chip Carrier (PLCC)	PRODUCT MATURITY/SATURATION(4)	-40 to +85 C	$3.95	
MC68HC705C8ACFS	44-Lead Ceramic Leaded Chip Carrier (CLCC)	PROD PHASE OUT/SEE LAST ORD DT(6) 30 Nov 2002	-40 to +85 C	$35.00	-
MC68HC705C8ACS	40-Pin Windowed Ceramic Dual In-Line Package (DIP)	PROD PHASE OUT/SEE LAST ORD DT(6) 30 Nov 2002	-40 to +85 C	-	
MC68HC705C8ACFB	44-Lead Quad Flat Pack (QFP)	PRODUCT MATURITY/SATURATION(4)	-40 to +85 C	$3.95	
MC68HC705C8ACB	42-Pin Shrink Dual In-Line Package (DIP)	PRODUCT MATURITY/SATURATION(4)	-40 to +85 C	$3.95	
KMC705C8ACB	Shrink DIP (70 mil spacing)	PRODUCT MATURITY/SATURATION(4)	-40 to +85 C	$3.95	
KMC705C8ACFB	10x10 mm Quad Flat Pack (QFP)	PRODUCT LAST SHIPMENTS(7)	-40 to +85 C	$3.95	
KMC705C8ACFN	Plastic Leaded Chip Carrier (PLCC)	PRODUCT MATURITY/SATURATION(4)	-40 to +85 C	$3.95	
KMC705C8ACFS	Ceramic Leaded - window (CLCC)	PROD PHASE OUT/SEE LAST ORD DT(6) 30 Nov 2002	-40 to +85 C	$35.00	-

KMC705C8ACP	Plastic Dual-in-Line (PDIP)	PRODUCT MATURITY/SATURATION(4)	-40 to +85 C	$3.95	
KMC705C8ACS	Shrink Plastic Dual-in-Line-window (SDIP)	PROD PHASE OUT/SEE LAST ORD DT(6) 30 Nov 2002	-40 to +85 C	$35.00	-
MC68HC705C8AB	Shrink DIP (70 mil spacing)	PRODUCT MATURITY/SATURATION(4)	0 to +70 C	$3.95	
MC68HC705C8ACFU	20x20 mm Plastic Quad Flat Pack (QFP)	PRODUCT MATURITY/SATURATION(4)	-40 to +85 C	$3.95	
MC68HC705C8ACPB	10x10 mm Low Profile Quad Flat Pack (LQFP)	PRODUCT MATURITY/SATURATION(4)	-40 to +85 C	$3.95	
MC68HC705C8AFN	Plastic Leaded Chip Carrier (PLCC)	PRODUCT MATURITY/SATURATION(4)	0 to +70 C	$3.95	
MC68HC705C8AFU	20x20 mm Plastic Quad Flat Pack (QFP)	PRODUCT MATURITY/SATURATION(4)	0 to +70 C	$3.95	
MC68HC705C8AVFN	Plastic Leaded Chip Carrier (PLCC)	PRODUCT MATURITY/SATURATION(4)	-40 to +105 C	$4.15	
MC68HSC705C8ACFN	Plastic Leaded Chip Carrier (PLCC)	PRODUCT MATURITY/SATURATION(4)	-40 to +85 C	$4.34	
MC68HSC705C8ACFS	Ceramic Leaded - window (CLCC)	REMOVED FROM ACTIVE PORTFOLIO(8)	-40 to +85 C	-	-
MC68HSC705C8ACP	Plastic Dual-in-Line (PDIP)	PRODUCT MATURITY/SATURATION(4)	-40 to +85 C	$4.34	
MC68HSC705C8ACS	Shrink Plastic Dual-in-Line-window (SDIP)	PROD PHASE OUT/SEE LAST ORD DT(6) 30 Nov 2002	-40 to +85 C	-	-
MC705C8ACFNR2	Plastic Leaded Chip Carrier (PLCC)	PRODUCT MATURITY/SATURATION(4)	-40 to +85 C	$4.06	
SPAKHC705C8ACFS	Ceramic Leaded - window (CLCC)	PRODUCT MATURITY/SATURATION(4)	-40 to +85 C	$82.66	
KMC705C8ACPB	-	PRODUCT MATURITY/SATURATION(4)	-	-	-
MC68HC705C8APB	-	PRODUCT MATURITY/SATURATION(4)	-	$5.60	
MCC68HC705C8A	-	PRODUCT MATURITY/SATURATION(4)	-	-	-
SPAKHSC705C8ACP	-	PRODUCT MATURITY/SATURATION(4)	-	-	-

[top]

Motorola Home | Semiconductors | Login | Support | Contact Us | Site Map
Products | Documentation | Tools | Design Resources | Applications

68HC705C8A : Microcontroller

The 68HC705C8a microcomputer (MCU) is a member of Motorola's M68HC05 family of low-cost single-chip microcomputers. This 8-bit MCU includes keyboard scanning logic, a high current pin, a COP watchdog timer, and read-only (ROM) security feature.

Block Diagram

68HC705C8A Features

- 68HC05 CPU
  - 8Kbytes EPROM
  - 304 bytes RAM
- Timer System
  - 16-bit Capture/Compare Timer system
  - Computer Operating Properly Watchdog Timer
- 24 Bidirectional I/O Lines and 7 Input-Only lines
- On-Chip Oscillator with Crystal/Ceramic Resonator
- Fully Static Operation
- Serial Communications
  - Asynchronous Serial Communications Interface (SCI)
  - Synchronous Serial Peripheral Interface (SPI)

Return to Top

68HC705C8A Documentation

Documentation

Application Note

ID	Name	Vendor ID	Format	Size K	Rev #	Date Last Modified	Order Availability
AN-HK-22	MC68HC05SR3 and MC68HC705SR3 Design Notes	MOTOROLA	pdf	0	0	1/01/1994	ORDER
AN-HK-23	MC6805R3 and MC68HC05SR3 Technical Comparison	MOTOROLA	pdf	0	0	1/01/1994	ORDER
AN1050_D	Designing for Electromagnetic Compatibility (EMC) with HCMOS Microcontrollers	MOTOROLA	pdf	82	0	1/01/2000	-
AN1055/D	M6805 16-Bit Support Macros	MOTOROLA	pdf	1048	0	1/01/1990	ORDER
AN1066/D	Interfacing the MC68HC05C5 SIOP to an I2C Peripheral	MOTOROLA	pdf	196	1	11/01/2001	ORDER
AN1067/D	Pulse Generation and Detection with Microcontroller Units	MOTOROLA	pdf	242	1	5/31/2002	ORDER
AN1212/D	J1850 Multiplex Bus Communication Using the MC68HC705C8 and the SC371016 J1850 Communications Interface (JCI)	MOTOROLA	pdf	377	1	11/13/2001	ORDER

AN1212SW	Software files for AN1212 zipped (exe file included)	MOTOROLA	zip	3329	0	1/01/1992	-
AN1222/D	Arithmetic Waveform Synthesis with the HC05/08 MCUs	MOTOROLA	pdf	24	0	1/01/1993	ORDER
AN1222SW	Software Files for AN1222 zipped	MOTOROLA	zip	20	0	1/01/1995	-
AN1226/D	Use of the 68HC705C8A in Place of a 68HC705C8	MOTOROLA	pdf	90	4	1/01/1996	ORDER
AN1227/D	Using 9346 Series Serial EEPROMs with 6805 Series Microcontrollers	MOTOROLA	pdf	401	1	1/01/1997	ORDER
AN1227SW	Software files zipped for AN1227	MOTOROLA	zip	151	1	1/01/1997	-
AN1228/D	Interfacing the M68HC05 MCU to the MC145051 A/D Converter	MOTOROLA	pdf	160	2	1/01/1997	ORDER
AN1228SW	Software for AN1228 zippedr	MOTOROLA	zip	27	2	1/01/1997	-
AN1256/D	Interfacing the HC05 MCU to a Multichannel Digital-to-Analog Converter Using the MC68HC705C8A and the MC68HC705J1A	MOTOROLA	pdf	89	1.1	1/01/1995	ORDER
AN1256SW	Software files for AN1256 zipped	MOTOROLA	zip	80	1.1	1/01/1995	-
AN1259/D	System Design and Layout Techniques for Noise Reduction in MCU-Based Systems	MOTOROLA	pdf	78	0	1/01/1995	ORDER
AN1262/D	Simple Real-Time Kernels for M68HC05 Microcontrollers	MOTOROLA	pdf	84	0	1/01/1995	ORDER
AN1262SW	Software files for AN1262	MOTOROLA	zip	11	0	1/01/1995	-
AN1263/D	Designing for Electromagnetic Compatibility with Single-Chip Microcontrollers	MOTOROLA	pdf	104	0	1/01/1995	ORDER
AN1292/D	Adding a Voice User Interface to M68HC05 Applications	MOTOROLA	pdf	155	0	1/01/1996	ORDER
AN1292SW	Software files for AN1292 zipped	MOTOROLA	zip	215	0	1/01/1996	-
AN1298/D	Variations in the Motorola MC68HC(7)05Cx Family	MOTOROLA	pdf	200	0	1/01/1996	ORDER
AN1516/D	Liquid Level Control Using a Motorola Pressure Sensor	MOTOROLA	pdf	77	2	1/24/2003	ORDER
AN1667/D	Software SCI Implementation to the MISC Communication Protocol	MOTOROLA	pdf	112	0	7/10/2002	ORDER
AN1667SW	Software for AN1667, zip format	MOTOROLA	zip	93	1.0	7/31/2002	-
AN1688/D	MISC Bus Slave Switch Node	MOTOROLA	pdf	243	0	7/11/2002	ORDER
AN1705/D	Noise Reduction Techniques for Microcontroller-Based Systems	MOTOROLA	pdf	67	0	1/01/1999	ORDER
AN1723/D	Interfacing MC68HC05 Microcontrollers to the IBM AT Keyboard Interface	MOTOROLA	pdf	274	0	1/01/1997	ORDER
AN1734/D	Pulse Width Modulation Using the 16-Bit Timer	MOTOROLA	pdf	102	0	1/01/1998	ORDER
AN1734SW	Software files for AN1734 zipped	MOTOROLA	zip	2	0	1/01/1997	-
AN1744/D	Resetting Microcontrollers During Power Transitions	MOTOROLA	pdf	80	0	1/01/1998	ORDER
AN1745/D	Interfacing the HC705C8A to an LCD Module	MOTOROLA	pdf	157	0	1/01/1998	ORDER
AN1745SW	Software files for AN1745 zipped	MOTOROLA	zip	3	0	1/01/1998	-

AN1752/D	Data Structures for 8-Bit Microcontrollers	MOTOROLA	pdf	213	1	5/07/2001	ORDER
AN1755/D	Interfacing the MC68HC705C8A to the DS2430A 256-Bit 1-Wire EEPROM	MOTOROLA	pdf	129	0	1/01/1998	ORDER
AN1757/D	Add a Unique Silicon Serial Number to the HC05	MOTOROLA	pdf	105	0	1/01/1998	ORDER
AN1758/D	Add Addressable Switches to the HC05	MOTOROLA	pdf	111	0	1/01/1998	ORDER
AN1761/D	Interfacing the MC68HC705C8A to the X76F041 PASS SecureFlash	MOTOROLA	pdf	179	0	1/01/1998	ORDER
AN1771/D	Precision Sine-Wave Tone Synthesis Using 8-Bit MCUs	MOTOROLA	pdf	250	0	1/01/1998	ORDER
AN1775/D	Expanding Digital Input with an A/D Converter	MOTOROLA	pdf	86	1	1/01/1998	ORDER
AN1818/D	Software SCI Routines with the 16-Bit Timer Module	MOTOROLA	pdf	84	0	1/01/1999	ORDER
AN1820/D	Software I2C Communications	MOTOROLA	pdf	55	0	1/01/1999	ORDER
AN1820SW	Software files for AN1820 zipped	MOTOROLA	zip	2	0	1/01/1998	-
AN2103/D	Local Interconnect Network (LIN) Demonstration	MOTOROLA	pdf	953	0	12/01/2000	ORDER
AN2159/D	Digital Direct Current Ignition System Using HC08 Microcontrollers	MOTOROLA	pdf	129	0	11/20/2001	ORDER
AN2159SW	AN2159SW	MOTOROLA	zip	182	1	3/08/2002	-
AN4006/D	Digital Captive Discharge Ignition System Using HC05/HC08 8-Bit Microcontrollers	MOTOROLA	pdf	61	0	3/27/2000	ORDER
AN442/D	Driving LCDs with M6805 Microprocessors	MOTOROLA	pdf	1134	0	1/01/1991	ORDER
AN463/D	68HC05K0 Infra-red Remote Control	MOTOROLA	pdf	111	0	1/01/1992	ORDER
AN464/D	Software Driver Routines for the Motorola MC68HC05 CAN Module	MOTOROLA	pdf	2859	0	1/01/1993	ORDER
AN477/D	Simple A/D for MCUs without Built-In A/D Converters	MOTOROLA	pdf	224	0	1/01/1993	ORDER
AN499/D	Let the MC68HC705 Program Itself	MOTOROLA	pdf	154	0	7/01/1996	ORDER
AN991/D	Using the Serial Peripheral Interface to Communicate Between Multiple Microcomputers	MOTOROLA	pdf	251	1	1/28/2002	ORDER
ANE416/D	MC68HC05B4 Radio Synthesizer	MOTOROLA	pdf	1958	0	1/01/1988	ORDER

Brochure

ID	Name	Vendor ID	Format	Size K	Rev #	Date Last Modified	Order Availability
BR68HC08FAMAM/D	68HC08 Family: High Performance and Flexibility	MOTOROLA	pdf	57	2	5/21/2003	ORDER
FLYREMBEDFLASH/D	Embedded Flash: Changing the Technology World for the Better	MOTOROLA	pdf	68	2	5/21/2003	ORDER

Data Sheets

ID	Name	Vendor ID	Format	Size K	Rev #	Date Last Modified	Order Availability
MC68HC705C8A/D	68HC705C8A Technical Data Book	MOTOROLA	pdf	2428	3	3/21/2002	ORDER

Engineering Bulletin

ID	Name	Vendor ID	Format	Size K	Rev #	Date Last Modified	Order Availability
EB166/D	System Design Considerations: Converting from the MC68HC805B6 to the MC68HC705B16 Microcontroller	MOTOROLA	pdf	757	0	1/01/1993	ORDER
EB180/D	Differences between the MC68HC705B16 and the MC68HC705B16N	MOTOROLA	pdf	20	0	1/01/1996	ORDER
EB181/D	Frequently Asked Questions and Answers for the M68HC05 Family MCAN Module	MOTOROLA	pdf	181	0	1/01/1997	ORDER
EB349/D	RAM Data Retention Considerations for Motorola Microcontrollers	MOTOROLA	pdf	45	1	6/22/2000	ORDER
EB396/D	Use of OSC2/XTAL as a Clock Output on Motorola Microcontrollers	MOTOROLA	pdf	49	0	6/19/2002	ORDER
EB413/D	Resetting MCUs	MOTOROLA	pdf	62	0	1/01/2000	ORDER
EB421/D	The Motorola MCAN Module	MOTOROLA	pdf	78	0	2/23/2000	ORDER

Errata - Click here for important errata information

ID	Name	Vendor ID	Format	Size K	Rev #	Date Last Modified	Order Availability
68HC705C8AMSE1/D	68HC705C8A Device Information Sheet: 0E20T-2E79R Mask Sets	MOTOROLA	pdf	11	1	11/06/1996	-
68HC705C8AMSE2/D	68HC705C8A Device Information Sheet: 1E20T-3E79R Mask Sets	MOTOROLA	pdf	14	1	1/01/1995	-
68HC705C8AMSE3/D	68HC705C8A Device Information Sheet: 0E20T-3E79R Mask Sets	MOTOROLA	pdf	17	3	7/16/2001	-

Fact Sheets

ID	Name	Vendor ID	Format	Size K	Rev #	Date Last Modified	Order Availability
CWDEVSTUDFACTHC08	Development Studio	MOTOROLA	pdf	48	2	5/13/2002	-

Product Change Notices

ID	Name	Vendor ID	Format	Size K	Rev #	Date Last Modified	Order Availability
PCN7701	QFP 10X10 ASSY MOVE FROM SHC TO BAT3	MOTOROLA	htm	16	-	7/09/2002	-
PCN7899	44/52/68 PLCC ASSY MOVE FROM SDI TO KLM	MOTOROLA	htm	31	0	8/14/2002	-
PCN7900	LQFP 10X10 ASSY MOVE FROM SDI TO BAT3	MOTOROLA	htm	16	0	8/14/2002	-
PCN7977	14X14 QFP ASSY MOVE FROM SDI TO KLM	MOTOROLA	htm	17	0	9/12/2002	-
PCN8103	10X10 LQFP ASSY MOVE FROM SHC TO BAT3	MOTOROLA	htm	16	0	10/08/2002	-
PCN8901	BINDING STRAP CHANGE FOR QFP PRODUCTS	MOTOROLA	htm	5	0	5/21/2003	-

Reference Manual

ID	Name	Vendor ID	Format	Size K	Rev #	Date Last Modified	Order Availability
M68HC05TB/D	HC05 Family - Understanding Small Microcontrollers	MOTOROLA	pdf	2866	2	1/01/1998	ORDER
MC68HC05CXRG/D	MC68HC05C4, C8, C9, MC68HC705C8, MC68HC805C4, MC68HCL05C4, C8, MC68HSC05C4, C8 Programming Reference	MOTOROLA	pdf	3150	1	2/23/2000	-

Selector Guide

ID	Name	Vendor ID	Format	Size K	Rev #	Date Last Modified	Order Availability
SG1002	Analog Selector Guide - Quarter 4, 2003	MOTOROLA	pdf	579	0	10/24/2003	ORDER
SG1006	Microcontrollers Selector Guide - Quarter 4, 2003	MOTOROLA	pdf	826	0	10/24/2003	ORDER
SG1010	Sensors Selector Guide - Quarter 4, 2003	MOTOROLA	pdf	219	0	10/24/2003	ORDER
SG1011	Software and Development Tools Selector Guide - Quarter 4, 2003	MOTOROLA	pdf	287	0	10/24/2003	ORDER
SG2000CR	Application Selector Guide Index and Cross-Reference.	MOTOROLA	pdf	95	3	11/11/2003	ORDER
SG2039	Application Selector Guide - Vacuum Cleaners Vacuum Cleaners	MOTOROLA	pdf	0	0	6/17/2003	ORDER

Return to Top

68HC705C8A Tools

Hardware Tools

Adapters

ID	Name	Vendor ID	Format	Size K	Rev #	Order Availability
PA44-P(Z)P	Prototyping Adapter	LOGSYS	-	-	-	-
PA705C8-DP(-PP)	Prototyping Adapter	LOGSYS	-	-	-	-
PA705C8-DX-QF	Prototyping Adapter	LOGSYS	-	-	-	-
PA705C8-PD	Programming Adapter	LOGSYS	-	-	-	-
PA705C8-QD-16	Programming Adapter	LOGSYS	-	-	-	-

Emulators/Probes/Wigglers

ID	Name	Vendor ID	Format	Size K	Rev #	Order Availability
AX-6811	AX-6811	HITEX	-	-	-	-
IC10000	iC1000 PowerEmulator	ISYS	-	-	-	-
IC20000	iC2000 PowerEmulator	ISYS	-	-	-	-
IC40000	iC4000 ActiveEmulator	ISYS	-	-	-	-

Evaluation/Development Boards and Systems

ID	Name	Vendor ID	Format	Size K	Rev #	Order Availability
KITMMDS05C	Modular Development System (MMDS) Kits	MOTOROLA	-	-	-	BUY
KITMMEVS05C	Modular Evaluation System (MMEVS)	MOTOROLA	-	-	-	BUY
M68CBL05B	Low-noise Flex Cable	MOTOROLA	-	-	-	BUY
M68CBL05C	Low-noise Flex Cable	MOTOROLA	-	-	-	BUY
M68EM05C9A	Emulation Module	MOTOROLA	-	-	-	BUY
M68ICS05C	In-Circuit Simulator for the 68HC05C and 68HC705C	MOTOROLA	-	-	-	BUY

Programmers

ID	Name	Vendor ID	Format	Size K	Rev #	Order Availability
MP8011A	Gang Programmer Base Unit	SOFTEC	-	-	-	-
POWERLAB	Universal Programmer	SYSGEN	-	-	-	-

Software

Application Software

Application Development Framework

ID	Name	Vendor ID	Format	Size K	Rev #	Order Availability
HC705C8AH	C header file for 68HC705C8A	MOTOROLA	zip	10	-	-

Code Examples

ID	Name	Vendor ID	Format	Size K	Rev #	Order Availability
C8THERMSW	HC05 Software Example: Home Thermostat example using the 705C8 with indoor/outdoor temperature and time of day	MOTOROLA	zip	11	-	-
FLOAT05COD	Floating Point routines	MOTOROLA	zip	13	-	-
HC05DELAYSW	HC05 Software Example: Subroutine that delays for a whole number of milliseconds	MOTOROLA	zip	2	-	-
HC05EXSW	Library containing software examples in assembly for 68HC05	MOTOROLA	zip	45	-	-
HC05KEYINTSW	HC05 Software Examples: Using keyboard interrupts and decoding a matrix keypad	MOTOROLA	zip	7	-	-
HC05KEYPADSW	HC05 Software Example: Keypad debounce and decode. When a key is found, it is changed to ASCII and displayed on an LCD	MOTOROLA	zip	2	-	-
HC05LCDSW	HC05 Software Example: Initializes an LCD and displays ABCDEF...S	MOTOROLA	zip	1	-	-
HC05SCISW	HC05 Software Example: Serial Communications Interface example	MOTOROLA	zip	1	-	-
HC05SPISW	HC05 Software Example: Serial Peripheral Interface example	MOTOROLA	zip	1	-	-
HC05SWITCHSW	HC05 Software Example: Simple program that reads the state of a switch on a general-purpose I/O pin and lights an LED based on the state of the switch	MOTOROLA	zip	1	-	-
HC05TIMERSW	HC05 Software Example: Using the 68HC05 16-bit Timer	MOTOROLA	zip	1	-	-
J1APWMSW	HC05 Software Example: Low frequency PWM example using the 68HC705J1A real-time interrupt and timer overflow interrupt	MOTOROLA	zip	1	-	-
J1AUARTSW	HC05 Software example: Software UART example that transmits and receives data on the 68HC705J1A	MOTOROLA	zip	2	-	-
K1THERMSW	HC05 Software Example: Thermometer project using the 68HC705K1	MOTOROLA	zip	5	-	-
MATH16ACOD	General Math routines	MOTOROLA	asm	4	-	-
MATHAGBCOD	General Math routines	MOTOROLA	asm	6	-	-
SAMPPROGCOD	Example routines	MOTOROLA	exe	35	-	-
THERM-CCOD	Thermometer example in C	MOTOROLA	zip	11	-	-

Software Tools

Assemblers

ID	Name	Vendor ID	Format	Size K	Rev #	Order Availability
ASHC5ASM	DOS based freeware assembler	MOTOROLA	arc	55	0	-
AX6805	AX6805 relocatable/absolute macro assembler for HC05	COSMIC	-	-	-	-

Compilers

ID	Name	Vendor ID	Format	Size K	Rev #	Order Availability
CWHC05	CodeWarrior Development Tools for HC05	METROWERKS	-	-	-	BUY
CX6805	CX6805 C Cross Compiler for HC05	COSMIC	-	-	-	-

Debuggers

ID	Name	Vendor ID	Format	Size K	Rev #	Order Availability
CWHC05	CodeWarrior Development Tools for HC05	METROWERKS	-	-	-	BUY
ZAP 6805 MMDS	ZAP 6805 MMDS Debugger	COSMIC	-	-	-	-
ZAP 6805 SIM	ZAP 6805 Simulator Debugger	COSMIC	-	-	-	-
AX-6811	AX-6811	HITEX	-	-	-	-

IDE (Integrated Development Environment)

ID	Name	Vendor ID	Format	Size K	Rev #	Order Availability
CWHC05	CodeWarrior Development Tools for HC05	METROWERKS	-	-	-	BUY
IDEA05	IDEA05 integrated development environment for HC05	COSMIC	-	-	-	-
IC-SW-OPR	winIDEA	ISYS	-	-	-	-

Models

Instruction Set Simulator

ID	Name	Vendor ID	Format	Size K	Rev #	Order Availability
PE68HC05SIM	Windows upgrades for P&E's simulator software for 68HC05	PEMICRO	html	0	-	-

Performance and Testing

ID	Name	Vendor ID	Format	Size K	Rev #	Order Availability
AX-6811	AX-6811	HITEX	-	-	-	-

Return to Top

Orderable Parts Information

PartNumber	Package Info	Tape and Reel	Life Cycle Description (code)	Budgetary Price QTY 1000+ ($US)	Additional Info	Order Availability
KMC705C8ACB	PSDIP 42	No	REMOVED FROM ACTIVE PORTFOLIO(8)	-	more	BUY
KMC705C8ACFN	PLCC 44	No	REMOVED FROM ACTIVE PORTFOLIO(8)	-	more	BUY
KMC705C8ACP	PDIP 40	No	REMOVED FROM ACTIVE PORTFOLIO(8)	-	more	BUY

KMC705C8ACPB	LQFP44 10*10*1.4P0.8	No	REMOVED FROM ACTIVE PORTFOLIO(8)	-	more	-
MC68HC705C8AB	PSDIP 42	No	PRODUCT MATURITY/SATURATION(4)	$3.95	more	BUY
MC68HC705C8ACB	PSDIP 42	No	PRODUCT MATURITY/SATURATION(4)	$3.95	more	BUY
MC68HC705C8ACFB	QFP 44 10*10*2.0P0.8	No	PRODUCT MATURITY/SATURATION(4)	$3.95	more	BUY
MC68HC705C8ACFN	PLCC 44	No	PRODUCT MATURITY/SATURATION(4)	$3.95	more	BUY
MC68HC705C8ACFU	QFP64 14*14*2.2P0.8	No	PRODUCT MATURITY/SATURATION(4)	$3.95	more	BUY
MC68HC705C8ACP	PDIP 40	No	PRODUCT MATURITY/SATURATION(4)	$3.95	more	BUY
MC68HC705C8ACPB	LQFP44 10*10*1.4P0.8	No	PRODUCT MATURITY/SATURATION(4)	$3.95	more	BUY
MC68HC705C8AFN	PLCC 44	No	PRODUCT MATURITY/SATURATION(4)	$3.95	more	BUY
MC68HC705C8AFU	QFP64 14*14*2.2P0.8	No	PRODUCT MATURITY/SATURATION(4)	$3.95	more	BUY
MC68HC705C8APB	LQFP 44 10*10*1.4P0.8	No	PRODUCT MATURITY/SATURATION(4)	$5.60	more	BUY
MC68HC705C8AVFB	QFP 44 10*10*2.0P0.8	No	PRODUCT MATURITY/SATURATION(4)	-	more	-
MC68HC705C8AVFN	PLCC 44	No	PRODUCT MATURITY/SATURATION(4)	$4.15	more	BUY
MC68HSC705C8ACFN	PLCC 44	No	PRODUCT MATURITY/SATURATION(4)	$4.34	more	BUY
MC68HSC705C8ACP	PDIP 40	No	PRODUCT MATURITY/SATURATION(4)	$4.34	more	BUY
MC705C8ACFNR2	PLCC 44	Yes	PRODUCT MATURITY/SATURATION(4)	$4.06	more	BUY
MCC68HC705C8A	CHIPS SM <50000 SQ MILS	No	PRODUCT MATURITY/SATURATION(4)	-	more	-
SPAKHC705C8ACFS	PDIP 40	No	REMOVED FROM ACTIVE PORTFOLIO(8)	-	more	-
SPAKHSC705C8ACP	PDIP 40	No	REMOVED FROM ACTIVE PORTFOLIO(8)	-	more	-

NOTE: Are you looking for an obsolete orderable part? Click **HERE** to check our distributors' inventory.

Return to Top

Related Links

- Automotive
- Microcontrollers
- Motor Control

Return to Top

